FORECASTING THE TURKISH YIELD CURVE USING DEEP LEARNING AND ECONOMETRIC MODELS: A COMPARATIVE ANALYSIS

EFE MERT USTAOĞLU

ÖZYEĞİN UNIVERSITY
JUNE, 2025

FORECASTING THE TURKISH YIELD CURVE USING DEEP LEARNING AND ECONOMETRIC MODELS: A COMPARATIVE ANALYSIS

by
Efe Mert Ustaoğlu

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Financial Engineering

Advisor: Asst. Prof. Emrah Ahi

Graduate School of Science and Engineering
Özyeğin University
İstanbul

June, 2025

FORECASTING THE TURKISH YIELD CURVE USING DEEP LEARNING AND ECONOMETRIC MODELS: A COMPARATIVE ANALYSIS

Approved by:

Asst. Prof. Emrah Ahi, Advisor
Department of International Finance
Özyeğin University

Asst. Prof. Levent Güntay
Department of International Finance
Özyeğin University

Asst. Prof. Rıza Ergün Arsal
Department of Logistics Management
Istanbul Bilgi University

Approval Date: May 13, 2025

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor, Dr. Emrah Ahi, for his invaluable support, guidance, and encouragement throughout this study. His insights and experience significantly contributed to the development and completion of this work.

I am especially thankful to my wife, Yasemin Ustaoğlu, whose patience, support, and unwavering belief in me have been my greatest source of strength during this challenging but rewarding journey.

DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Efe Mert Ustaoğlu

ABSTRACT

This thesis investigates the modeling and forecasting of the Turkish Treasury bond yield curve using both traditional time series models and deep learning approaches. The study focuses on evaluating the performance of models such as Autoregressive (AR), Random Walk (RW), the Dynamic Nelson-Siegel (DLNS) model with fixed and time-varying decay parameters, and Long Short-Term Memory (LSTM) networks. Daily zero-coupon yield data between July 2012 and February 2025 were used, covering maturities ranging from 3 months to 10 years. Forecasting performance was assessed using an out-of-sample expanding window framework for different horizons. Root Mean Squared Error (RMSE) was employed as the main accuracy metric, and the Diebold-Mariano (DM) test was applied to determine whether performance differences across models were statistically significant. The empirical findings show that while simpler models such as AR and RW perform well in the short term, their forecasting accuracy deteriorates as the horizon increases. In contrast, LSTM models—particularly those trained on latent factors derived from the Nelson-Siegel framework—demonstrate superior generalization performance in medium- and long-term forecasts. Notably, the DLNS model with a time-varying λ also outperforms benchmark models at longer horizons. These results highlight the value of incorporating structured factor dynamics and deep learning techniques for forecasting complex financial time series such as the yield curve.

Keywords: Yield Curve, Deep Learning, Nelson-Siegel Model, LSTM, Time Series Forecasting

ÖZET

Bu tez, Türkiye Hazine tahvili getiri eğrisinin hem geleneksel zaman serisi modelleri hem de derin öğrenme yaklaşımlarıyla modellenmesini ve tahmin edilmesini incelemektedir. Çalışmada; Otoregresif (AR), Random Walk (RW), sabit ve zamanla değişen sönümleme parametrelerine sahip Dinamik Nelson-Siegel (DLNS) modeli ile Uzun Kısa Süreli Bellek (LSTM) ağları gibi modellerin tahmin performansı değerlendirilmektedir. Analizde, 3 aydan 10 yıla kadar vadeleri kapsayan ve Temmuz 2012 ile Şubat 2025 tarihleri arasındaki günlük sıfır kupon getiri verileri kullanılmıştır. Modellerin tahmin performansı, farklı vadeler için dış örneklem genişleyen pencere yöntemiyle değerlendirilmiştir. Tahmin doğruluğu için temel ölçüt olarak Karekök Ortalama Hata (RMSE) kullanılmış, modeller arası performans farklarının istatistiksel olarak anlamlı olup olmadığını test etmek amacıyla Diebold-Mariano testi uygulanmıştır. Ampirik bulgular, AR ve RW gibi basit modellerin kısa vadede başarılı sonuçlar verdiğini ancak tahmin ufku uzadıkça performanslarının belirgin şekilde azaldığını göstermektedir. Buna karşılık, özellikle Nelson-Siegel çerçevesinden türetilmiş latent faktörlerle eğitilen LSTM modelleri, orta ve uzun vadeli tahminlerde üstün genelleme performansı sergilemiştir. Ayrıca, zamanla değişen λ parametresine sahip DLNS modeli de uzun vadede referans modellere kıyasla daha başarılı sonuçlar üretmiştir. Bu bulgular, yapılandırılmış faktör dinamiklerinin ve derin öğrenme tekniklerinin, getiri eğrisi gibi karmaşık finansal zaman serilerinin tahmininde sağladığı önemli katkıyı ortaya koymaktadır.

Anahtar Kelimeler: Getiri Eğrisi, Derin Öğrenme, Nelson-Siegel Modeli, LSTM, Zaman Serisi Tahmini

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AR	Autoregressive
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
BIST	Borsa Istanbul
BPTT	Backpropagation Through Time
BVAL	Bloomberg Valuation Service
CBRT	Central Bank of the Republic of Turkey
CIR	Cox-Ingersoll-Ross
CNN	Convolutional Neural Networks
DL	Deep Learning
DLNS	Deep Learning Nelson-Siegel
DM	Diebold-Mariano
DNS	Dynamic Nelson-Siegel
ELU	Exponential Linear Unit
FADNS	Factor-Augmented Dynamic Nelson-Siegel
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GRU	Gated Recurrent Unit
HJM	Heath-Jarrow-Morton
IQR	Interquartile Range
LSTM	Long Short-Term Memory

MAE Mean Absolute Error

ML Machine Learning

MSE Mean Squared Error

NS Nelson-Siegel

NSS Nelson-Siegel-Svensson

OLS Ordinary Least Squares

PCA Principal Component Analysis

PReLU Parametric Rectified Linear Unit

ReLU Rectified Linear Unit

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RW Random Walk

SGD Stochastic Gradient Descent

VAR Vector Autoregression

YTM Yield to Maturity

1. INTRODUCTION

Understanding the dynamics of the yield curve is essential for financial decision-making, monetary policy design, and risk management in both public and private sectors. The yield curve, which depicts the relationship between bond yields and their maturities, reflects critical market expectations about interest rates, inflation, and economic activity. Due to its interpretive power, it has become a widely used analytical tool in monetary policy assessments (BIS, 2005 [1]) and financial market analysis. As such, accurate modeling and forecasting of the yield curve are vital for policymakers, investors, and financial institutions.

Given the importance and complexity of forecasting the yield curve, a wide array of methods and modeling frameworks have been developed over time. Among these, the pioneering contribution of Diebold and Li (2006) transformed the static Nelson-Siegel (1987) model into a dynamic structure (DNS), enabling it to be applied within a time series forecasting context. By estimating the Nelson-Siegel parameters using an econometric approach, they demonstrated that the resulting forecasts outperformed those generated by naïve models. This work underscored the importance of forecasting based on latent factors derived from the Nelson-Siegel framework and laid the foundation for numerous extensions and innovations in the literature.

Following this approach, various researchers have sought to enhance the DNS model's forecasting performance. One of the most notable extensions comes from Koopman et al. (2010), who proposed a more flexible version of the DNS model by treating the decay parameter (λ) as a fourth latent factor. This extended dynamic model demonstrated superior in-sample fit and improved forecasting performance relative to fixed-parameter alternatives. Building on the core structure of the DNS model, Fernandes and Vieira (2019) introduced forward-looking macroeconomic variables, resulting in the Factor-Augmented Dynamic Nelson-Siegel (FADNS) model. Their empirical findings show that the augmented model performs particularly well at longer horizons, outperforming traditional autoregressive benchmarks. Overall, the findings suggest that the DNS model remains a versatile and expandable framework for term structure modeling and forecasting.

This thesis investigates whether deep learning methods can improve yield curve forecasting by modeling the Nelson-Siegel parameters. Accordingly, the study develops and compares several forecasting approaches, including traditional models such as Autoregressive (AR) and Random Walk (RW), and dynamic deep learning-based models under the DLNS framework (Dynamic Nelson-Siegel with fixed or time-varying λ). In addition, several Long Short-Term Memory (LSTM) architectures are explored, including models trained directly on the latent factor representations.

Using daily zero-coupon yield data from July 2012 to February 2025, these models are evaluated within an out-of-sample expanding window framework. Forecast accuracy is measured using Root Mean Squared Error (RMSE), and model comparisons are conducted through the Diebold-Mariano (DM) test. The empirical results reveal that while simpler models such as AR and RW perform reasonably well in the short term, their accuracy deteriorates at longer forecasting horizons. In contrast, LSTM based models that are especially based on the Nelson-Siegel factor structure demonstrate improved predictive accuracy and more stable long-term dynamics. These findings reveal that integrating Nelson-Siegel factors with modern deep learning architectures has the potential to enhance the accuracy of yield curve forecasts.

2. FUNDAMENTALS OF THE YIELD CURVE: CONCEPTS AND MODELS

2.1 Importance of the Yield Curve

The time value of money is the principle that money received today is more valuable than the same amount received in the future, due to its potential to generate returns over time. This concept plays a significant role in financial decision-making for economic agents. Investment, spending, and saving decisions are based on an assessment of the benefits that money can provide over time. Thus, measuring the present and future utility of money is crucial to analyzing financial decisions. The interest rate serves as the unit of measurement for the time value of money. Therefore, understanding the relationship between interest rates and time is one of the key issues in finance.

The yield curve is a graph that illustrates the relationship between the yield rates of bonds or debt instruments with different maturities and the time remaining until maturity. It is one of the most important tools used in an economy to understand the time value of money. Typically, the horizontal axis represents the maturity periods, while the vertical axis represents the yield rates. When constructing the yield curve, it is assumed that the financial instruments used have similar risk profiles and that the only factor determining the yield is the maturity.

The yield curve is usually derived from government debt instruments. Due to the high liquidity of these instruments and the assumption of zero credit risk, changes in the yield curve are considered to depend solely on maturity. These characteristics make government bonds and treasury bills ideal tools for estimating the yield curve. If government debt instruments were available for every maturity, the graph of their interest rates would directly form the yield curve. However, since such instruments are not traded for every maturity, yields for some maturities cannot be observed. Therefore, the yield curve is modeled to estimate yields for maturities where no trading occurs, based on the yields of existing maturities. The resulting yield curve helps determine the value of debt instruments across different maturities. At the same time, this curve forms the basis for pricing many financial and real assets.

The yield curve serves as a crucial reference point for market participants, providing significant insights into economic expectations, inflation, risk perception, and monetary policy. The relative value of short-term yields compared to long-term yields reflects the expected trajectory of future short-term interest rates. As such, the slope of the yield curve serves as a key indicator of economic activity. A steepening yield curve implies that economic activity is likely to accelerate, and inflation may rise. Conversely, a flattening or inverted yield curve signals an economic slowdown and suggests that inflation may decrease. When long-term yields rise relative to short-term yields, it indicates that the market views the economic outlook more positively.

The yield curve also holds significant importance for economic policy. The impact of short-term interest rates on economic indicators operates through the monetary transmission mechanism. While central banks determine short-term interest rates, long-term interest rates are shaped by market participants' expectations. Central banks use yield curve estimations to gauge investors' reactions to policy decisions and to obtain empirical representations of interest rate term structures [2]. Consequently, the yield curve is a useful tool for analyzing the effects of policy decisions and designing new ones. Moreover, the yield curve is also one of the primary tools used in planning public sector borrowing. Just like investment and borrowing decisions in the private sector, fiscal policy also relies on this instrument in its decision-making processes.

2.2 Modeling the Yield Curve

The yield curve should not include effects caused by extreme fluctuations in the price of a particular security that do not reflect general market expectations. The relationship between maturity and yield should be isolated from factors such as credit risk, liquidity risk, and other types of risk. Therefore, the nature of the financial instruments from which the curve is derived is of significant importance.

Yield curves are typically constructed using the prices of government bonds with different maturities that are actively traded in the secondary market. The fixed-income nature of these securities—with predetermined maturities and payment schedules—facilitates the analysis of the relationship between yield and time. Due to their high liquidity and the availability of prices across various maturities, government securities are frequently used in modeling the yield curve.

Generally, government securities with maturities longer than one year are referred to as government bonds, while those with maturities shorter than one year are called treasury bills. The government securities used in yield curve construction can vary in structure. For instance, some issued instruments may be zero-coupon and sold at a discount, while others may offer coupon payments with varying payment schedules.

To evaluate debt instruments of different structures together in a consistent framework, a standardized representation is required. For this purpose, the yields corresponding to the discount function of each instrument are computed.

The price of a risk-free bond with a known maturity and coupon payments can be expressed as follows. In this representation, credit risk is considered negligible, and the discount rate is determined solely by inflation expectations and the real interest rate.

$$P_t = \sum_{m=1}^{M} \frac{C}{(1+s_{t,m})^m} + \frac{N}{(1+s_{t,M})^M} \tag{2.1}$$

If a fixed-income financial instrument were a zero-coupon bond, the calculation would be straightforward. The discount rate of such an instrument could be easily derived using the time to maturity and the final redemption value. The resulting discount rate represents the cost of a risk-free debt instrument with a maturity of m and a repayment amount of N. The market price of the fixed-income instrument in the secondary market can be considered its present value. Based on this observed market price, the discount rate can be calculated and interpreted as the annualized risk-free yield until maturity.

The rate that equates the present value of a fixed-income asset to the sum of its future cash flows is called the Yield to Maturity (YTM). If a fixed-income security is held to maturity, and no intermediate trading occurs, the discount rate equals the YTM.

However, fixed-income government securities traded in the market come in various structures. Longer-term instruments typically involve coupon payments. Due to the presence of multiple cash flows, the calculation becomes more complex for coupon-bearing securities. Coupon-bearing bonds are fixed-income instruments with maturities longer than one year that make periodic payments. These payments can be structured in several ways: coupons may be paid at regular intervals, the principal may be repaid at maturity, or payments may be made in lump sums (balloon payments). Additionally, coupon amounts may be fixed as a proportion of the principal or vary based on a metric such as inflation.

In summary, depending on the structure of the debt instrument, both the timing and size of future cash flows can vary. It is still possible to estimate a Yield to Maturity (YTM) that equates these cash flows to the current market price. This estimation is based on the assumption that future cash flows are correctly anticipated and that all received coupon payments are reinvested at the same YTM rate until maturity. However, since YTM provides a single average rate, it does not reveal differences across maturities and therefore is insufficient for constructing the full yield curve. A proper yield curve must reflect not only current interest rates but also expectations about future interest rate movements. YTM alone does not convey forward-looking rate dynamics. For this reason, when deriving the spot yield curve for various maturities, two primary methods are employed: coupon stripping and bootstrapping.

Yield curve bootstrapping is a technique used to calculate the spot interest rates of zero-coupon bonds. Since most bonds in the market are coupon-bearing, spot rates for zero-coupon instruments are not directly observable. The bootstrap method enables the derivation of spot rates step-by-step using the prices of coupon-bearing bonds. Starting with the interest rates of short-term instruments, the rates of longer-term instruments can be deduced. This indirect method allows interest rates to be estimated for each maturity point.

Coupon stripping refers to separating each coupon payment and the principal of a bond into individual securities. In other words, each cash flow of a bond is isolated and transformed into a zero-coupon bond. This enables investors to hold zero-coupon securities with fixed cash flows for specific maturities. As a result, markets become more liquid and efficient. This method also plays an important role in the accurate calculation of the yield curve.

Spot rates are the market discount rates of risk-free zero-coupon bonds and are often referred to as zero rates. When bond prices are calculated using a set of spot rates, an arbitrage-free price is obtained.

The par rate is the market discount rate for a specific maturity that allows a bond to be priced at par (i.e., its nominal value). Par rates are derived from the spot rates up to that maturity.

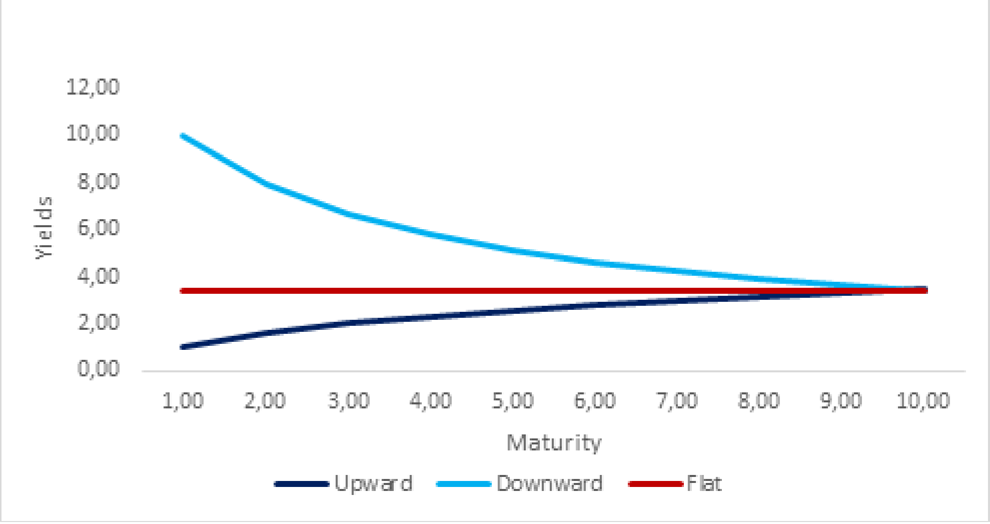
The forward curve represents interest rates (i.e., forward rates) that begin at a specific point in the future and continue thereafter. These rates are also derived from spot rates and are calculated using formulas that establish the relationship between different spot rates. Forward rates reflect the marginal return an investor can earn by reinvesting at a future date and indicate market expectations regarding future interest rates.

Since both par rates and forward rates are derived from spot rates, the shape of the spot yield curve is closely linked to the shape of the par and forward curves. If interest rates are increasing, par rates are usually lower than spot rates, while forward rates are higher than spot rates. Conversely, if interest rates are declining, these relationships are reversed. These dynamics play a key role in bond pricing and interest rate forecasting.

Yield curves are typically upward-sloping, meaning that long-term instruments offer higher returns than short-term ones. This reflects the additional risk and opportunity cost associated with holding long-term investments. Since long-term investors must wait longer for returns, a higher yield compensates for the increased uncertainty.

Figure 2.1 illustrates the various shapes that a yield curve may take. Depending on market movements (shocks), the yield curve may shift upward or downward, and its slope and curvature may change. A parallel shift implies a change in the overall level of yields across all maturities. When short-term yields change more than long-term yields, the slope of the curve is affected. The curvature, on the other hand, is mainly influenced by changes in medium-term yields, resulting in a more hump-shaped curve.

Figure 2.1: *Yield Curve Shapes*



2.3 Theories for Yield Curve Modeling

Due to the limited availability of pure discount bonds in the market, it is not possible to directly obtain zero-coupon interest rates for every maturity. In order to derive the zero rates that form the yield curve, we rely on other government bonds that offer coupon payments. The estimation of these zero rates requires the use of specific models and numerical techniques. This is known as the term-structure estimation problem. The estimated zero rates obtained through these models form the yield curve, which is expected to best replicate the market prices of government bonds. Accordingly, the performance of the estimated curve is evaluated based on how accurately it reflects the prices of these securities in the market.

The yield curve, which shows the relationship between yield and maturity, represents a complex estimation problem with many unknowns. Over time, numerous methods have been developed to construct the yield curve. While some of these methods are more complex than others, there is no universally accepted or superior model. Each model offers a unique perspective and presents different approaches and methodologies depending on the intended application.

For example, if the primary goal is to accurately price illiquid government bonds or derivatives, smoothness may not be a key criterion when selecting a model. However, if the goal is to analyze general interest rate expectations of economic agents using the zero-coupon yield curve, a smoother curve may be preferable. Still, excessive smoothness—or the lack thereof—can lead to the loss of important economic information. Therefore, it is crucial to strike a balance between goodness of fit and smoothness, depending on the application.

As a result, different models may be used for different purposes, and in many cases, multiple models may be employed to evaluate a problem from various perspectives. For instance, central banks utilize zero-coupon and forward rate curves for different analytical and policy objectives, making it necessary to rely on a variety of modeling approaches [3].

Although the proposed models cannot be strictly classified, certain categorizations have been suggested in the literature. Anderson et al. (1996) [4] classified yield curve modeling approaches into two main categories: asset pricing models based on equilibrium and arbitrage principles, and statistical models derived from observed asset prices. Over time, many variations of these models have been developed, creating an extensive body of literature. The purpose of estimating the yield curve, the pool of financial instruments used for the estimation, and the structural features of the economy can all influence the preference for one model over another. Understanding the strengths and limitations of each model helps in selecting the most suitable one for a given purpose, thereby enhancing the performance of the yield curve estimation.

In the following sections of this study, the main theories developed for modeling the yield curve will be discussed. First, the traditional theories, which aim to explain the relationship between maturity and yield, will be presented. Then, modern approaches, which offer more mathematically rigorous frameworks, will be examined. Finally, the core focus of this thesis—the Nelson-Siegel approach—will be analyzed in detail.

2.3.1 Conventional Theories

Conventional models primarily focus on the term structure of the yield curve rather than offering mathematical solutions. These theories provide logical interpretations of the shape and structure of the yield curve, rather than deriving it through formal modeling techniques.

Conventional theories are generally categorized under the following main frameworks: the liquidity preference theory, the pure expectations theory, the segmented markets hypothesis, and the preferred habitat theory.

2.3.1.1 Liquidity Preference Theory

This theory explains the term structure of interest rates by arguing that long-term investments are riskier than short-term ones, and therefore, risk-averse investors demand a higher premium for holding longer-term bonds. At its core, the theory is based on the principle that money received today is preferred over money received in the future. Within this framework, long-term bonds are considered less desirable compared to short-term bonds, which leads to the idea that long-term bonds should offer a liquidity premium to compensate for their lower preference. As a result, the theory typically implies an upward-sloping yield curve. However, the theory falls short in explaining inverted yield curves and changes in curvature across different maturities.

2.3.1.2 Pure Expectations Theory

This theory suggests that forward rates are unbiased predictors of future spot rates. In other words, long-term interest rates are composed of a series of expected short-term interest rates. Under the assumption of a risk-neutral world, investors are indifferent between investing in a long-term bond or successively rolling over short-term bonds, as both strategies are expected to yield the same return. The theory also implies that long-term interest rates are equal to the average of expected future short-term rates. However, empirical studies have shown that the implied spot rates derived from market data are often overstated, suggesting the possible existence of a risk premium between short- and long-term investments. Despite this limitation, the model provides a valuable framework for explaining the behavior of interest rates, particularly in the short-term segment of the yield curve.

2.3.1.3 Segmented Markets Hypothesis

According to the Segmented Markets Hypothesis, interest rates at different maturities on the yield curve are determined independently by the supply and demand dynamics within each maturity segment. This theory posits that there are distinct investor groups for short-term, intermediate-term, and long-term securities. The presence of maturity constraints among various institutional investors—such as fund managers, insurance companies, pension funds, and money market funds—supports this view. These investors typically operate within preferred maturity ranges based on their specific investment mandates. As a result, the shape of the yield curve is influenced by the relative supply and demand within each segment, rather than by expectations about future interest rates. The main shortcoming of this hypothesis is that it disregards the relationship between yields of different maturities, offering no unified framework for modeling the entire term structure of interest rates.

2.3.1.4 Preferred Habitat Theory

Similar to the Segmented Markets Hypothesis, the Preferred Habitat Theory suggests that investors have preferences for specific maturity segments, and that yields across the yield curve are determined by supply and demand within those segments. However, unlike the Segmented Markets Hypothesis, this theory allows for flexibility: if yields in other maturities become sufficiently attractive, investors are willing to shift away from their preferred maturities to take advantage of those opportunities. By acknowledging both investors' maturity preferences and their expectations about the future shape of the yield curve, the Preferred Habitat Theory offers a more comprehensive explanation of the term structure compared to the segmented markets approach.

2.3.2 *Modern Theories*

Models categorized as modern theories offer the ability to perform quantitative analysis and aim to derive the yield curve using various mathematical and econometric techniques. Term structure models are used to describe how interest rates evolve over time and play a fundamental role in the pricing of bonds and other fixed-income securities. These models can be broadly classified into different categories depending on their structure, underlying assumptions, and modeling approach. Although there is no universally accepted classification in the literature, it is common to analyze them under distinct groups based on their framework. Modern models will be analyzed under two headings: structural models and statistical models.

2.3.2.1 Structural Models

These models explain the term structure of interest rates based on underlying economic factors and assumptions. They focus on modeling the internal dynamics of interest rates and forecasting their expected future behavior. Structural models are typically divided into two main categories: Equilibrium Models (Endogenous Models) and No-Arbitrage Models (Exogenous Models).

Equilibrium models are grounded in economic theory and describe how interest rates evolve within an equilibrium framework. They generally assume that markets are in equilibrium and that interest rates are determined by fundamental factors such as the supply and demand for funds. These models are used to forecast the future movements of interest rates and are often applied in contexts such as bond pricing.

Unlike no-arbitrage models, equilibrium models are not designed to fit the existing term structure of interest rates observed in the market. Instead, they are based on assumptions of economic equilibrium and aim to explain the long-term behavior of interest rates. Equilibrium models can be classified as either single-factor or multi-factor models. Their simplicity and analytical tractability have contributed to their widespread use. However, their inability to fully match market data and the potential for producing unrealistic results in some cases are important limitations to consider.

These models typically consist of two main components: drift and volatility. The deterministic part of the model is represented by the drift component ($\lambda\, dt$), while the stochastic part is represented by the volatility term ($\sigma\, dW$), where dW denotes a Wiener process. A generalized structural model can be expressed by the following stochastic differential equation:

$$dr = \lambda\, dt + \sigma\, dW, \quad dW \sim N(0, \sqrt{dt}) \tag{2.2}$$

In structural models, the drift component (λ) can generally be categorized into three types: driftless, constant drift, and time-varying drift. Considering factors such as parallel shifts in the yield curve, risk premiums, and convexity effects, the time-dependency of the drift term becomes particularly important for the model's functionality. In many structural models, the drift component is indeed time-dependent, and various models have been developed to describe how this component evolves over time.

One of the earliest and most well-known structural models is the Vasicek Model (1977), which assumes that interest rates exhibit mean-reverting behavior [5]. The model uses stochastic differential equations to describe the dynamics of short-term interest rates, suggesting that interest rates tend to revert toward a long-term average. In this framework, the speed at which the short rate reverts to the long-term mean is determined by the coefficient in the drift term. This mean-reversion mechanism acts as a control against excessively high or low interest rates. However, the Vasicek model has some notable limitations—it allows for negative interest rates and relies on constant parameters, which may reduce its realism in certain market conditions.

An extension of the Vasicek model is the Cox-Ingersoll-Ross (CIR) Model, which addresses the issue of negative interest rates by employing a square-root diffusion process [6]. Unlike the Vasicek model, the CIR model ensures that interest rates remain non-negative. In this model, volatility is proportional to the square root of the interest rate level, meaning that as interest rates decline, volatility also decreases. This feature aligns more closely with actual market behavior, especially during periods of low interest rates.

Another structural model is the Dothan Model [7], which models interest rates as following a geometric Brownian motion. The Dothan model assumes that interest rates are lognormally distributed, which inherently prevents the possibility of negative rates. Unlike the Vasicek model, the Dothan model does not incorporate mean reversion, limiting its ability to accurately capture the long-term behavior of interest rates.

The Longstaff-Schwartz Model (1992) is a two-factor equilibrium model designed to more realistically capture the dynamics of interest rates [8]. It simultaneously considers both the short-term interest rate and the volatility of interest rates, using two state variables. This framework allows for a more flexible representation of both the level and the shape of the yield curve. In this model, volatility is modeled independently of the interest rate level, which provides a closer fit to real-world financial market conditions.

Despite their theoretical appeal, a key drawback of equilibrium models is their inability to perfectly fit the current yield curve. This limitation can pose significant challenges in applications such as bond pricing, where precise alignment with market prices is essential.

No-arbitrage models are based on the fundamental assumption that arbitrage opportunities do not exist in financial markets. These models are widely used to describe the dynamics of interest rates and are specifically designed to fit the observed term structure of interest rates in the market. In such models, parameter calibration is performed in a way that ensures the interest rates generated by the model match the interest rates observed in the market.

Interest rates in no-arbitrage models are represented using stochastic processes, and these models can be developed as either single-factor or multi-factor frameworks. One of the earliest and most influential examples of this class is the Ho-Lee Model (1986), which was the first to apply a no-arbitrage approach to modeling the evolution of interest rates over time [9]. The Ho-Lee model incorporates a time-dependent drift term and assumes that interest rates follow a normal distribution. However, because the model allows for negative interest rates, it can yield unrealistic results, especially in low-rate environments. While it successfully fits market data, the model does not include the property of mean reversion, which is considered a limitation in certain applications.

The Hull-White Model is an extension of the Ho-Lee model that addresses this issue by introducing mean reversion into the framework [10]. Despite this improvement, it still allows for negative interest rates, which may pose practical limitations in real-world applications. The Black-Derman-Toy Model (1990) offers a different approach by assuming that interest rates follow a lognormal distribution, effectively avoiding the problem of negative rates [11]. The model represents the evolution of interest rates using a binomial tree structure, which maps out the possible future values of interest rates and the probabilities associated with reaching those values. This tree-based approach facilitates the pricing of interest rate derivatives by providing a discrete framework for modeling future uncertainty.

Another prominent no-arbitrage model is the Heath-Jarrow-Morton (HJM), introduced in 1992. What distinguishes the HJM model from earlier models is that it directly models the forward rate curve rather than the short-term interest rate itself [12]. Compared to the Hull-White model, the HJM framework is more flexible and is particularly well-suited for applications such as pricing interest rate derivatives and risk management.

To better capture the complexities of interest rate movements, especially over longer horizons and under more volatile conditions, multi-factor no-arbitrage models have also been developed. These models allow for a richer and more realistic representation of the term structure dynamics, improving both the accuracy of derivative pricing and the robustness of risk assessments in fixed-income markets.

2.3.2.2 Statistical Models

Statistical models are data-driven approaches used to estimate and model the yield curve based on directly observed market data. These models are also commonly referred to as curve-fitting techniques. Due to their technical nature, statistical models generally have limited capacity to explain underlying economic factors, as they are primarily focused on empirical accuracy rather than theoretical interpretation. Unlike structural models, statistical approaches do not rely on assumptions about the internal dynamics of interest rates. Instead, their primary objective is to estimate the curve that best fits the observed yield data in the market. Statistical models are typically classified into two main categories: Spline Models and Parsimonious Parametric Models. Spline models aim to capture local variations in the yield curve by fitting piecewise polynomial functions, while parsimonious models use a limited number of parameters to describe the overall shape of the yield curve with a smooth functional form. The key advantage of statistical models lies in their flexibility and ability to closely match market data. However, their lack of theoretical grounding can be a disadvantage in applications that require interpretation of interest rate movements based on economic fundamentals.

Spline models are among the first statistical models used in the literature.Spline models construct the yield curve using polynomials that connect predefined knot points, defining a different polynomial for each segment of the data and ensuring smooth transitions between segments. The primary goal of spline models is to guarantee the continuity of the curve and its derivatives, resulting in a smooth and accurate fit to the observed data.

Various types of spline techniques exist, including Linear Spline, Cubic Spline, B-Spline, and Smoothing Spline. Since a separate polynomial is defined for each data interval, spline models are particularly well-suited for capturing complex curve shapes. This flexibility is among their key advantages. However, the use of too many segments may lead to overfitting, especially if the number and placement of knot points are not carefully chosen. In such cases, the resulting curve may fit the noise in the data rather than the underlying trend, potentially leading to misleading conclusions.

One of the earliest applications of spline models in yield curve estimation was proposed by McCulloch (1971) [13]. His approach aimed to model the yield curve flexibly using piecewise polynomial functions, recognizing the limitations of classical methods—such as straight-line fitting or simple polynomials—in accurately capturing the actual shape of the yield curve. McCulloch introduced a cubic spline-based framework that allowed for greater adaptability to market data.

Building upon this approach, Vasicek and Fong (1982) developed a model that further improved the smoothness and accuracy of the yield curve estimation [14]. Unlike polynomial splines, their model employed exponential functions as spline segments, which yielded more reliable results, especially over longer maturities.

Another notable contribution came from Fisher et al. [15], who introduced a practical framework for modeling the zero-coupon yield curve using the natural cubic spline method. A significant innovation in their approach was the inclusion of a penalty function to prevent excessive roughness in the fitted curve, enhancing both smoothness and stability.

One of the most important subcategories of statistical models is parsimonious parametric models. These models emerged as an alternative to spline-based piecewise smooth curve models, which often involve a large number of parameters and complex structures. While spline models can offer great flexibility, their complexity can make them difficult to interpret. Parsimonious models, on the other hand, focus on capturing the fundamental relationships in the data using a simpler and more interpretable structure. As a result, the outcomes are easier to explain, and the risk of overfitting to the training data is significantly reduced.

The foundational work that established this class of models was the Nelson-Siegel (NS) model, developed by Charles Nelson and Andrew Siegel in 1987. The Nelson-Siegel model expresses the yield curve in a mathematically compact form and uses a small number of parameters to capture the curve's shape. The model is capable of representing the short-, medium-, and long-term dynamics of the yield curve effectively.

The Nelson-Siegel model represents the yield to maturity as a function of time to maturity using the following equation:

$$\gamma(\tau) = \beta_0 + \beta_1 \left(\frac{1 - e^{-\lambda\tau}}{\lambda\tau} \right) + \beta_2 \left(\frac{1 - e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau} \right) \tag{2.3}$$

Where:

- $\gamma(\tau)$: yield to maturity for a given time to maturity,
- τ: time to maturity,
- $\beta_0, \beta_1, \beta_2$: model parameters,
- λ: decay factor.

In this model, the parameter β_0 represents the long-term level of interest rates. The parameter β_1 reflects the slope of the yield curve, specifically the difference between short-term and long-term rates. The parameter β_2 captures the curvature, which describes how medium-term interest rates behave relative to short- and long-term rates.

The factor loadings in the model determine how the impact of each component (β_0, β_1, β_2) changes depending on the time to maturity. The loading of β_0 is constant and always equal to 1, meaning it affects all maturities equally. The loading of β_1 is a function that declines with maturity—it decreases rapidly at first and then more gradually—capturing the steepness of the yield curve. In contrast, the loading of β_2 initially increases and then declines, meaning the curvature factor has the most significant influence in medium-term maturities.

The value of the decay parameter λ determines at which maturities the effects of the slope and curvature become most apparent. A high value of λ implies that the effects of β_1 and β_2 emerge at shorter maturities, while a low value of λ pushes these effects further out into longer maturities.

Overall, the Nelson-Siegel model has been widely adopted due to its interpretability, flexibility, and ability to capture key features of the yield curve using just a few parameters. It forms the foundation of many extensions and is still actively used by central banks and financial institutions in yield curve modeling and forecasting.

2.4 Yield Curve Modeling and Forecasting with the Nelson-Siegel Model

The level, slope, and curvature factors provide a powerful framework for explaining the movements of the yield curve of fixed-income securities. Litterman and Scheinkman (1991) used Principal Component Analysis (PCA) to analyze variations in bond yields and demonstrated that the vast majority of the variability—over 90%—can be explained by these three key factors [16].

Building upon the Nelson-Siegel (1987) framework, Svensson (1994) proposed an extension known as the Nelson-Siegel-Svensson (NSS) model. In his work, Lars E.O. Svensson increased the flexibility of the original model, enhancing its ability to capture more complex yield curve shapes by introducing an additional curvature term. This improvement allowed the model to better adapt to the empirical characteristics of the term structure observed in financial markets.

The extended version, known as the NSS model, is expressed as follows:

$$\gamma(\tau) = \beta_0 + \beta_1 \left(\frac{1 - e^{-\lambda\tau}}{\lambda\tau} \right) + \beta_2 \left(\frac{1 - e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau} \right) + \beta_3 \left(\frac{1 - e^{-\lambda_2\tau}}{\lambda_2\tau} - e^{-\lambda_2\tau} \right) \quad (2.4)$$

In this specification, the parameter β_3 acts as the second curvature component. By incorporating two distinct decay factors, λ_1 and λ_2, the model gains enhanced flexibility, allowing it to capture a wider variety of yield curve shapes. These decay factors govern the behavior of the yield curve across different maturity horizons: λ_1 generally influences the short- and medium-term sections of the curve, while λ_2 enables the model to account for long-term curvature and more intricate patterns in yield dynamics.

Overall, the introduction of these two decay parameters significantly increases the model's adaptability, enabling a more accurate and robust representation of the term structure under varying market conditions and economic scenarios.

Gürkaynak et al. [17] argued that the Svensson extension is often preferable to the original Nelson-Siegel model, as the slope of the yield curve tends to decline over longer maturities, necessitating the inclusion of a second curvature factor for improved accuracy. Similarly, Nymand-Andersen (2018) showed that the Svensson model demonstrates greater flexibility and goodness of fit compared to the Nelson-Siegel model. Söderlind and Svensson [**?**] also noted that while the original Nelson-Siegel model generally provides a sufficiently good fit, the Svensson model tends to outperform it in cases involving complex term structures.

Svensson's extended framework has been widely adopted by central banks and financial institutions for analyzing and forecasting the term structure of interest rates [18]. As a result, monetary policy decisions and risk management strategies are often based on this more robust modeling foundation.Various extensions of the Nelson-Siegel model have been proposed in the literature. For instance, Björk and Christensen (1999) introduced a second slope factor to the NS and NSS models [19]. However, Diebold et al. [20]

found that this additional factor contributed only marginally to the model's performance, concluding that a simpler model with fewer factors was generally sufficient. Additionally, De Pooter [21] highlighted the risk that a fifth factor could introduce problematic instability into the model.

On the other hand, Diebold et al. [22] proposed a reduced-form Nelson-Siegel model that excluded the curvature factor, arguing that the level and slope factors alone explained the vast majority of the variation in interest rates. However, empirical results showed that this two-factor version did not perform as effectively in forecasting as the full model.

In a pivotal contribution, Diebold and Li [23] proposed that the level, slope, and curvature of the yield curve should be allowed to evolve over time, estimating the parameters β_0, β_1 and β_2 using a dynamic framework. This approach, known as the Dynamic Nelson-Siegel (DNS), incorporates autoregressive processes to model the time-varying behavior of the parameters. Building on this framework, subsequent studies developed Factor-Augmented Dynamic Nelson-Siegel (FADNS) models that integrate macroeconomic variables into the term structure modeling process [22, 24].

Another important development was introduced by Christensen, Diebold, and Rudebusch (2009), who expanded the Nelson-Siegel model into an arbitrage-free framework, significantly enhancing its utility for modeling and forecasting interest rate dynamics [25].

Further innovation came from Nyholm (2015), who proposed the Reparametrized Dynamic Nelson-Siegel (RDNS) Model, a dynamic version of the classical NS model [26]. While the original NS model's parameters are technically interpreted as level, slope, and curvature, the RDNS model reparametrizes these into short-term interest rate, slope, and curvature, allowing for improved economic interpretability and better integration with macroeconomic variables.

The dynamic approach introduced by Diebold and Li has also played a foundational role in addressing the lack of studies focusing on forecasting the yield curve. In their framework, the decay factor λ is assumed to be fixed, while the other parameters are estimated using Ordinary Least Squares (OLS) regression.

In contrast, Koopman et al. (2010) extended the dynamic model by treating the decay factor (λ) as a fourth latent factor, estimating it through a vector autoregressive (VAR) structure [27].However, modeling the decay factor dynamically introduces a non-linear optimization problem. This challenge arises due to the existence of multiple local minima, sensitivity to small changes in asset prices, and the influence of different initial values on the solution produced by the optimization algorithm. These factors make the estimation of such models computationally complex and sensitive.

To address these difficulties, recent studies have explored artificial intelligence-based methods for parameter estimation. For instance, Gimeno and Nave [28] proposed a novel methodology using genetic algorithms to forecast the yield curve. Similarly, Gilli et al. [29] suggested the use of global optimization techniques to estimate the parameters of the Nelson-Siegel-Svensson (NSS) model, improving its estimation performance.

More recently, Sang-Heon Lee (2023) demonstrated that dynamic estimation of Nelson-Siegel parameters using sequential deep learning models such as LSTM and RNN leads to significantly better forecasting performance compared to naive models [30].

3. DEEP LEARNING MODELS

Financial forecasting is a critical process aimed at understanding market uncertainties and anticipating future trends. In the world of finance, forecasting methods underpin a wide range of areas, including investment decision-making, risk management, portfolio optimization, macroeconomic analysis, and central bank policies. Therefore, financial forecasting holds substantial importance not only for individual and institutional investors but also for governments, financial institutions, and corporations.

Traditional time series forecasting techniques include statistical models such as Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and Vector Autoregression (VAR). These models typically operate under certain assumptions and perform well in capturing linear relationships within the data. However, financial markets are inherently complex, nonlinear, and highly volatile, and they often exhibit non-stationary behavior from a statistical perspective. As a result, traditional methods may not always offer sufficiently accurate forecasts. In particular, external factors such as global economic events, macroeconomic indicators, central bank decisions, market sentiment, and news flows can cause significant fluctuations in asset prices. Modeling such variability requires more flexible and powerful forecasting techniques.

In recent years, Machine Learning (ML) and Deep Learning (DL) techniques have begun to outperform traditional models in identifying complex patterns and modeling nonlinear relationships in financial markets. In this context, the importance of financial time series forecasting continues to grow. Accurate forecasting models contribute to the development of investment strategies, the identification of market anomalies, and the improvement of risk management processes. As understanding the uncertainties inherent in financial markets and building effective strategies become more critical, the use of data science and artificial intelligence-based methods has become indispensable.

Machine learning refers to a class of models that enable computer systems to learn patterns from data and make predictions without being explicitly programmed. Unlike traditional programming, where a computer is given a fixed set of rules, machine learning algorithms create their own models by analyzing large datasets and use those models to make future predictions. Machine learning techniques are generally categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the model is trained on input-output pairs and learns the relationship between them to make predictions on new data. In unsupervised learning, the model is trained only on input data without corresponding outputs. Regression models are common examples of supervised learning, while Principal Component Analysis (PCA) is an example of unsupervised learning. In reinforcement learning, an agent interacts with an environment and learns through a system of rewards and penalties, gradually improving its decision-making process.

DL is a subfield of machine learning built on Artificial Neural Networks (ANN). While traditional ML algorithms often require manual feature selection, deep learning methods automatically learn high-level features from raw data through multi-layered neural networks. Therefore, DL models have been applied across a wide range of domains, including image, speech, video, and audio processing, as well as natural language understanding tasks such as topic classification, sentiment analysis, question answering, and language translation [31].

In the financial domain, deep learning represents a machine learning approach based on multi-layered artificial neural networks that can extract complex patterns and relationships from large datasets. Going beyond traditional statistical models and basic machine learning algorithms, deep learning has proven effective in fields such as time series forecasting, natural language processing, computer vision, and financial modeling. The literature includes various deep learning architectures, including Recurrent Neural

Network (RNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory networks.

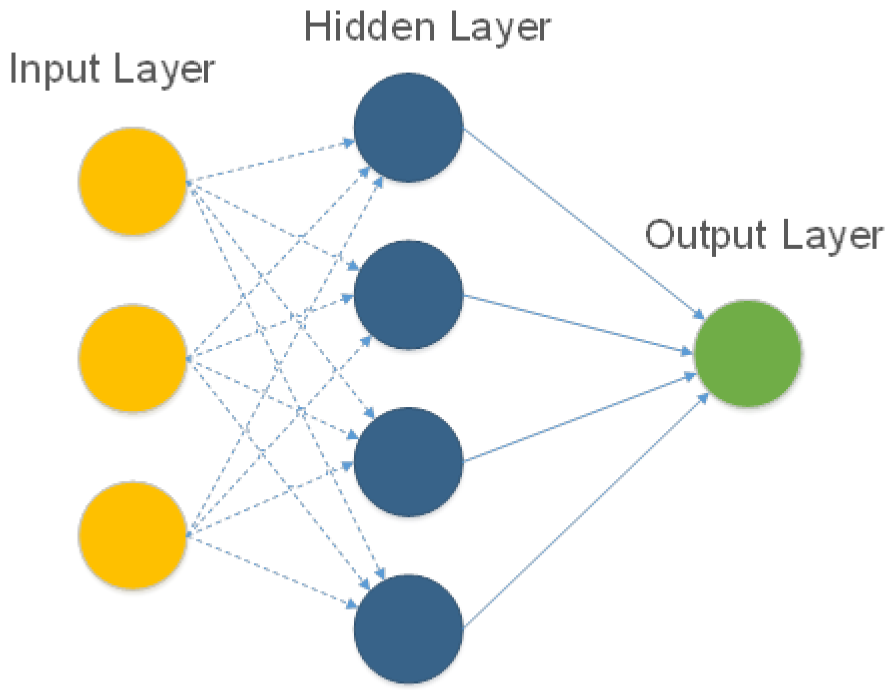
3.1 Artificial Neural Networks (ANN)

Artificial intelligence refers to computer systems capable of learning from data, adapting to environmental changes, and making autonomous decisions in pursuit of a specific goal. The fundamental objective of artificial intelligence is to develop systems that can think in a manner similar to human intelligence and interact with their environment. The artificial neural cell model of McCulloch and Pitts was introduced in their 1943 paper titled *"A Logical Calculus of the Ideas Immanent in Nervous Activity."* [32]. This work provided a mathematical framework for the basic functioning principles of neural systems and inspired the development of artificial neural networks. The introduction of the backpropagation algorithm in the 1980s enabled the training of multi-layer networks [33]. Since the 2000s, deep learning techniques and architectures such as convolutional and recurrent neural networks have revolutionized the field of machine learning [34].

ANN, inspired by biological neural networks, consist of interconnected input, hidden, and output layers comprising artificial neurons. In an artificial neural network, each neuron multiplies the incoming inputs by certain weights, adds a bias term, and passes the result through an activation function to produce an output.

Using weights and activation functions, the input data are processed within the network, enabling the learning of nonlinear relationships. Activation functions are mathematical transformation functions that determine how the network responds to specific inputs. Without activation functions, a neural network would behave merely as a linear model and fail to perform deep learning. Mathematically, the output of a neuron is calculated as follows:

Figure 3.1: *ANN Model Framework*



In artificial neural networks, each layer transforms the input using a weighted sum followed by an activation function. The weighted sum for the j-th neuron in layer l is given by:

$$z_j^l = \sum_i w_{ji}^l a_i^{l-1} + b_j^l \tag{3.1}$$

$$a_j^l = \sigma(z_j^l) \tag{3.2}$$

In this formula, w represents the weights, b the bias term, and σ the activation function. The process begins with feeding the input data into the network. Each neuron multiplies the inputs by their respective weights and sums the result with the bias. The resulting value is then passed through the activation function. This process continues until the output layer is reached. The prediction produced by the output layer is compared with the actual value to calculate the error. An example of a loss function used at the output layer is the Mean Squared Error (MSE), shown below:

$$E = \frac{1}{2} \sum_j (a_j^L - y_j)^2 \tag{3.3}$$

The difference between the output and the expected result y_j is calculated at the output layer. Subsequently, ANN models are optimized using the backpropagation algorithm. The backpropagation process involves the propagation of the error signal, computed at the output layer, backward through the preceding layers using the chain rule. This method enables the calculation of the gradients required to update the model's weights in a way that minimizes the difference between the output and the target.

First, the error term for the neuron in the output layer is defined as:

$$\delta_j^L = (a_j^L - y_j) \cdot \sigma'(z_j^L) \tag{3.4}$$

Here, a_j^L is the output of the neuron, y_j is the true value, and $\sigma'(z_j^L)$ is the derivative of the activation function. This formula is used to compute the error signal in the output layer. Then, the error terms in the hidden layers are calculated by applying the chain rule as follows:

$$\delta^l = \left(W^{(l+1)}\right)^T \delta^{(l+1)} \cdot \sigma'(z^l) \tag{3.5}$$

In this equation, $W^{(l+1)}$ is the weight matrix for layer $l+1$, $\delta^{(l+1)}$ is the error vector from the next layer, and $\sigma'(z^l)$ is the derivative of the activation function in layer l. This step ensures that the error of each neuron is proportionally related to the error signal from the next layer via the weights.

Then, based on the calculated error terms, the gradients for each weight and bias are obtained:

$$\frac{\partial E}{\partial w_{ji}^l} = a_i^{l-1} \cdot \delta_j^l \tag{3.6}$$

$$\frac{\partial E}{\partial b_j^l} = \delta_j^l \tag{3.7}$$

Finally, these gradients are used to update the weights and biases using optimization algorithms. The general update rules are as follows:

$$w_{ji}^l \leftarrow w_{ji}^l - \eta \cdot \frac{\partial E}{\partial w_{ji}^l} \tag{3.8}$$

$$b_j^l \leftarrow b_j^l - \eta \cdot \frac{\partial E}{\partial b_j^l} \tag{3.9}$$

Here, η represents the learning rate. These updates are repeated until the error signal is sufficiently propagated backward and the model performance reaches the desired level. This approach ensures error minimization and performance optimization during the training process of the neural network, thus facilitating the learning process.

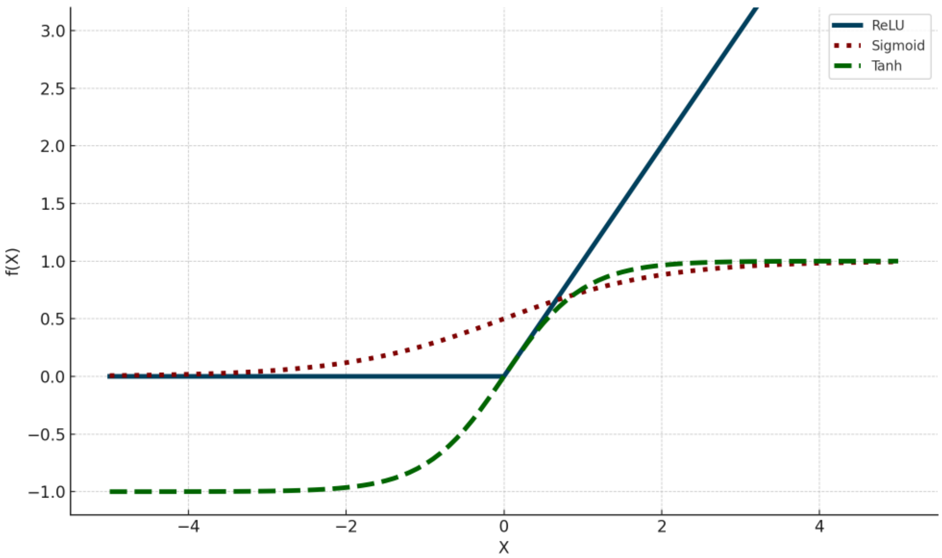
3.1.1 Hyperparameters

One of the key factors that directly affects model performance is its hyperparameters. Hyperparameters include the number of hidden layers, the number of units in each layer, regularization techniques such as dropout, L1, and L2, the initialization methods of weights (e.g., He or Xavier initialization), the choice of activation functions, learning rate, decay rate, momentum value, number of epochs, batch size, and the selected optimization algorithms. Various techniques such as Manual Search, Grid Search, Random Search, and Bayesian optimization can be used to determine the most suitable values for these hyperparameters. Additionally, building models through trial and error is also a widely used approach. Familiarity with specific models and datasets may help develop an intuitive skill in designing model architectures. It can be said that building deep learning models is partly an art, not purely a science. However, it should be emphasized that proper hyperparameter selection and optimization play a critical role in the overall success of the model.

3.1.1.1 Activation Functions

Activation functions are among the most critical components that determine the learning capacity of artificial neural networks. ReLU, Sigmoid, and Tanh are the most commonly used activation functions. In general, activation functions are categorized as either linear or non-linear. Due to the inherent limitations of linear functions, non-linear activation functions are typically preferred in practice.

Figure 3.2: *Activation Functions*



The sigmoid activation function is a non-linear function that compresses input values into a range between 0 and 1. This property makes it particularly useful in models that produce probability-based outputs. The curve of the sigmoid function has an S-shape (sigmoid curve) and exhibits different behaviors depending on the magnitude of the input values. For negative inputs, the output is close to 0, while for positive inputs, the output approaches 1. When the input value is 0, the output is exactly 0.5. This behavior helps neural networks direct values above or below a certain threshold toward specific classes. The sigmoid function is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.10}$$

The sigmoid activation function is well-suited for applications such as binary classification and logistic regression. However, in deep neural networks, it can lead to the vanishing gradient problem. For this reason, more modern activation functions such as ReLU and its variants are generally preferred in practice.

ReLU (Rectified Linear Unit) is one of the most widely used activation functions in artificial neural networks. Due to its simple structure and computational efficiency, it offers significant advantages in deep learning models. The mathematical formula of the ReLU function is as follows:

$$f(x) = \max(0, x) \tag{3.11}$$

This function outputs zero for negative input values, and passes positive input values directly as output. This simple yet effective structure enables neural networks to learn non-linear relationships while reducing computational cost and largely mitigating the vanishing gradient problem. One of the major advantages of this activation method is that, unlike sigmoid and *tanh*, it does not require exponential operations, thereby reducing computational complexity for large datasets. While the derivatives of sigmoid and *tanh* can become very small, slowing down learning, the derivative of ReLU is always 1 for positive inputs, which accelerates the learning process.

However, since ReLU sets negative inputs to zero, some neurons may become completely inactive. This situation, in which certain neurons effectively drop out of the model, is known as the *dead neuron problem*. To address this issue, improved variants such as Leaky Rectified Linear Unit (ReLU), Parametric Rectified Linear Unit (PReLU), and Exponential Linear Unit (ELU) have been developed.

The Tanh (Hyperbolic Tangent) activation function is a non-linear, zero-centered function that scales input values to a range between -1 and 1. The mathematical formula of the Tanh function is given as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.12}$$

The output range of the Tanh activation function lies between -1 and 1. For negative inputs, the output approaches -1; for positive inputs, it approaches 1. This property helps normalize inputs around zero, which leads to more balanced weight updates and accelerates the learning process.

Compared to the sigmoid function, Tanh is generally more advantageous for hidden layer activations because the gradient magnitudes tend to be larger, which facilitates a faster backpropagation process. However, when used in very deep neural networks, it can still suffer from the vanishing gradient problem. Additionally, for very large or very small input values, the derivative of the function approaches zero, which may slow down weight updates during backpropagation.

3.1.1.2 Model Training

During the optimization process, various algorithms are employed to update the weights. For instance, Stochastic Gradient Descent (SGD) operates on randomly selected mini-batches rather than the entire dataset at each iteration, thereby reducing computational cost. Momentum-based SGD accelerates the optimization by incorporating previous updates into the current direction. AdaGrad aims to improve the convergence performance of standard SGD by adapting the learning rate for each parameter individually.

RMSProp adjusts the learning rate for each parameter by dividing it by the moving average of the squared gradients specific to that parameter.

In addition, the ADAM algorithm combines the advantages of both momentum and RMSProp by incorporating estimates of the first and second moments of the gradients. This makes it particularly effective in online and non-stationary environments.

During backpropagation, if the error signal diminishes as it propagates backward through the layers and becomes too small by the time it reaches the earlier layers, the vanishing gradient problem occurs. This leads to insufficient weight updates in the early layers. Conversely, if the gradients grow excessively during backpropagation and cause numerical instability, the exploding gradient problem arises. This issue is particularly prominent in training deep neural networks or RNNs.

To mitigate this problem, a technique called gradient clipping is commonly applied. In this method, gradients are restricted from exceeding a specified threshold value, thereby preventing overly large updates and promoting stability during training.

3.1.1.3 Other Hyperparameters

In addition to the core hyperparameters previously discussed, there are several other hyperparameters that can influence the training dynamics and overall performance of neural networks. These include the choice of learning rate schedules, gradient clipping thresholds, weight decay parameters, early stopping criteria, and batch normalization usage. The selection and fine-tuning of these hyperparameters often depend on the complexity of the dataset, the depth of the network, and the specific goals of the modeling task. The number of hidden layers and the number of units in each layer define the architecture and representational capacity of the neural network. A deeper network with more hidden layers can learn more complex patterns, but it also increases the risk of overfitting and computational cost. Similarly, the number of units per layer controls how much information each layer can capture.

Regularization techniques, such as dropout, L1, and L2 regularization, are applied to prevent overfitting. Dropout randomly deactivates a subset of neurons during training, reducing dependency on specific neurons. L1 regularization adds a penalty based on the absolute values of weights, encouraging sparsity in the network, while L2 regularization penalizes the squared values of weights to limit their magnitude.

The initialization of weights is another critical factor. Poor initialization may cause gradients to vanish or explode during training. Techniques like He initialization and Xavier initialization are commonly used to maintain the variance of activations across layers, facilitating stable and efficient training

The learning rate is one of the most influential hyperparameters. It controls the size of the weight updates during gradient descent. A learning rate that is too high may cause the model to diverge, while a very small learning rate may slow down convergence significantly. To address this, a decay rate may be applied to gradually reduce the learning rate over time.

The number of epochs refers to how many times the entire training dataset is passed through the model, and the batch size determines the number of samples used in one forward/backward pass. Smaller batch sizes provide more frequent updates and can improve generalization, while larger batch sizes benefit from more stable gradient estimates.

It is important to note that the hyperparameters discussed above represent only a subset of the many possible configurations available in deep learning models. There exist many other hyperparameters that can significantly affect model performance depending on the specific task, data structure, and network architecture.

3.2 Convolutional Neural Networks (CNNs)

CNNs are deep learning architectures that demonstrate superior performance particularly in processing high-dimensional data such as images, videos, and similar formats. Unlike traditional artificial neural networks, CNNs possess the ability to automatically extract meaningful features from raw data. This capability not only enhances computational efficiency but also improves the model's generalization performance.

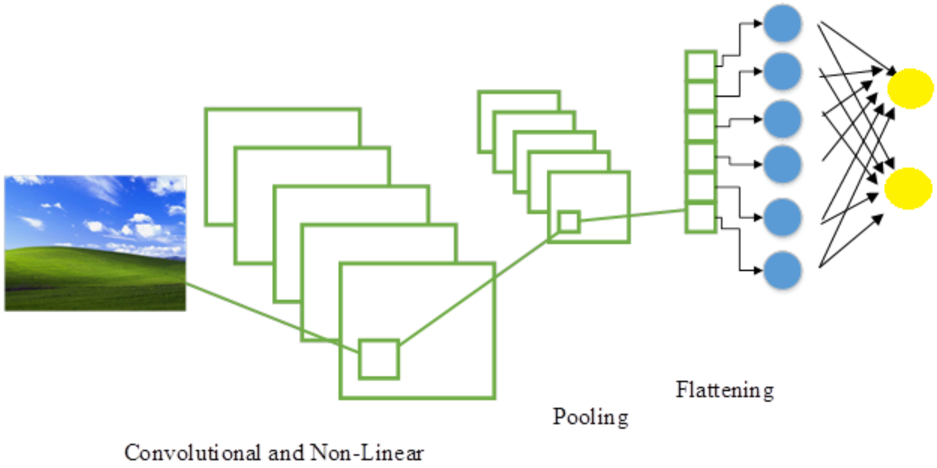
Although CNNs are predominantly used in the field of image processing, they can also be highly effective in tasks such as time series forecasting due to their strong feature extraction capabilities. By using moving windows, CNNs are able to detect specific patterns within a time series and utilize these features in the later stages of deep learning models.

The convolution operation is carried out by sliding small-sized filters (also known as kernels) over the input data. Each filter captures specific patterns (such as edges, corners, or textures) in local regions of the input. The resulting feature maps provide a condensed representation of the input data. In the initial layers, the convolutional layer is responsible for learning low-level features, while deeper layers enable the extraction of increasingly abstract representations.

Pooling layers reduce the dimensionality of the feature maps obtained from the convolutional layers. The most commonly used pooling method is max pooling, in which the maximum value within a defined window is selected, thereby reducing the spatial dimensions. Pooling not only decreases the computational load of the model but also promotes spatial invariance, allowing the model to be less sensitive to small spatial transformations in the input data.

The summarized features extracted by the convolutional and pooling layers are then passed to fully connected layers, where they are integrated. These layers constitute the final stage of the network, performing tasks such as classification or regression. The

Figure 3.3: *Overview of CNN Workflow*



nodes in the final layers of the network generate the final output based on the combined representation of the learned features.

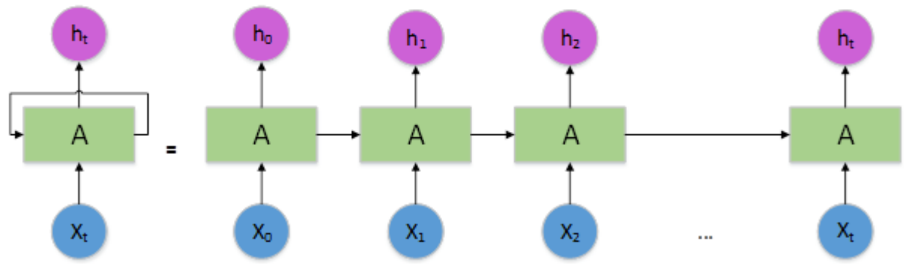
3.3 Recurrent Neural Networks (RNNs)

RNNs are deep learning architectures equipped with a dynamic memory structure, widely used for modeling sequential data types such as time series, text, speech, and video. Unlike traditional neural networks, RNNs are capable of learning temporal dependencies within data by carrying information from one time step to the next. The core structure of RNNs involves a recurrent loop that combines the input data at each time step with the previous hidden state to compute the current hidden state:

$$h_t = \varphi(W_{xh}\, x_t + W_{hh}\, h_{t-1} + b_h) \tag{3.13}$$

Here, x_t denotes the input at time step t, h_{t-1} is the hidden state from the previous time step, W_{xh} and W_{hh} represent the weight matrices for the input and hidden state connections, b_h is the bias term for the hidden state, and φ denotes the activation function.

Figure 3.4: *Recurrent Neural Network (RNN) Architecture*



The training of RNNs is carried out using Backpropagation Through Time (BPTT), which is an extension of the traditional backpropagation algorithm applied along the temporal dimension. In this method, error signals are propagated backward through successive time steps. However, learning long-term dependencies can be challenging due to the frequent occurrence of the vanishing and exploding gradient problems. These issues hinder the model's ability to effectively retain and update information from earlier time steps.

To address these limitations, advanced RNN architectures such as the LSTM model and the Gated Recurrent Unit (GRU) have been developed. More recently, Transformer-based models have also been introduced to overcome these challenges and further enhance the performance of sequential modeling tasks.

3.4 Long Short-Term Memory (LSTM)

LSTM is a type of artificial neural network specifically designed to work with time series and sequential data. Traditional RNN often struggle to learn long-term dependencies, a challenge commonly referred to as the vanishing gradient problem. LSTM addresses this issue through a specially designed architecture that allows it to learn and retain long-range dependencies more effectively. The LSTM model was introduced by

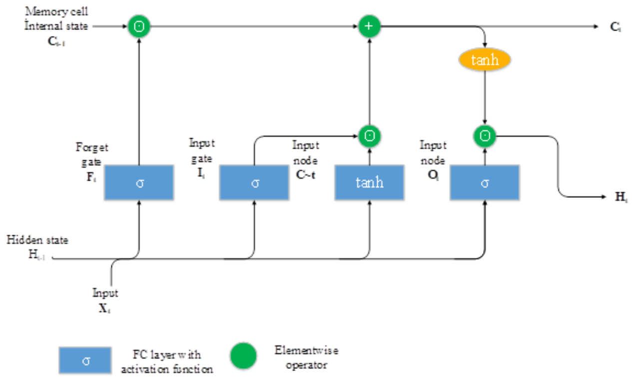
Sepp Hochreiter and Jürgen Schmidhuber in 1997 [35]. At that time, the limitations of RNNs—particularly their failure to maintain information over extended time intervals—led researchers to seek alternative solutions. By introducing innovative components such as memory cells and gating mechanisms, LSTM significantly improved the performance of traditional RNNs.

Unlike conventional RNNs, LSTM networks consist of memory cells and three main gates within each unit: the input gate, forget gate, and output gate. These gates regulate the flow of information and determine which information should be stored, forgotten, or passed on to the next step. The input gate decides how much of the new information should be added to the memory cell.

To model long-term dependencies in sequential data, LSTM uses gate mechanisms that control the cell state. Through these mechanisms, the model can selectively remember or forget information from the past, allowing it to effectively capture both short- and long-term dependencies. Each LSTM cell includes three principal gates in addition to the classic RNN cell structure. The forget gate determines which parts of the cell state should be discarded. It calculates a value using an activation function and compares it with a threshold to decide whether the information should be retained or forgotten. The input gate operates in two steps: first, it evaluates which new information should be added to the memory cell; second, it updates the cell state by combining this new information with the retained content from the forget gate. Finally, the output gate determines the hidden state, which is the output that will be passed to the next LSTM unit.

LSTM networks, through their gating mechanisms, can effectively model long-term dependencies by controlling which information should be forgotten, which should be retained, and how the cell state should be updated. Therefore, the most significant advantage of LSTM is its ability to learn long-range temporal dependencies.

Figure 3.5: *LSTM Architecture*



It is also possible to combine LSTM models with other deep learning architectures. For example, CNN-LSTM models, which integrate two different deep learning structures—CNN and LSTM—combine the strengths of both: feature extraction and temporal modeling. While the CNN component is responsible for extracting meaningful features from the input data, the LSTM component captures the temporal evolution of these features over time.

Moreover, traditional models used in interest rate forecasting, such as the DNS model, can also be enhanced by integrating them with deep learning-based models. Such hybrid approaches can lead to more accurate and stable predictions. In this study, the aim is to combine the capabilities of deep learning models with the conceptual framework of the Nelson-Siegel (NS) model to develop an effective and robust yield curve modeling approach.

4. ANALYSIS AND RESULTS

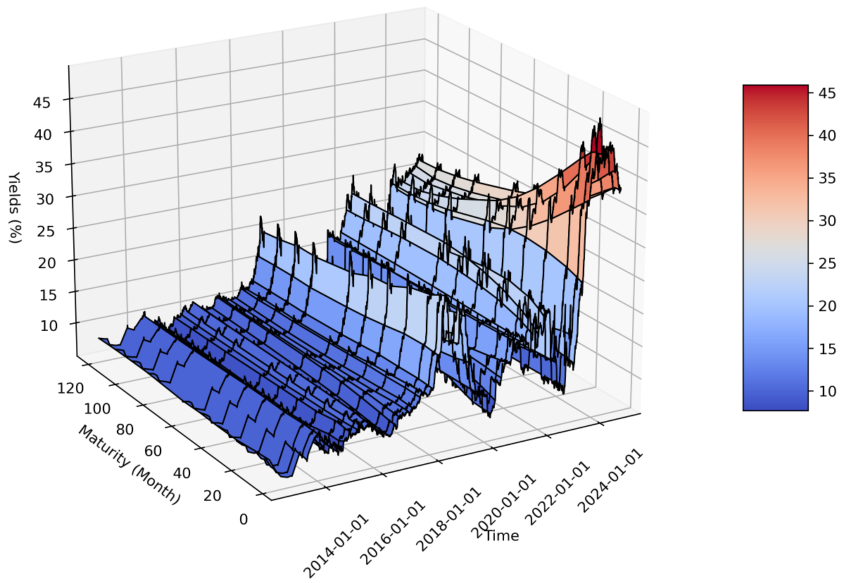
This section provides a detailed overview of the dataset used in the study, the methodologies applied, and the results obtained. The primary objective of the study is to forecast the Turkish Treasury bond yield curve using various time series forecasting methods. The forecasts were generated using different models and techniques, including LSTM, AR models, and the Random Walk (RW) model. By comparing the performance of these different model structures, the study aims to identify which method delivers the best forecasting performance.

4.1 Dataset

In this study, daily spot yield data of Turkish Treasury bonds between July 5, 2012, and February 20, 2025 were utilized. The analysis focuses on bond yields with maturities of 3, 6, 12, 24, 36, 48, 60, 84, 96, 108, and 120 months. The dataset was constructed to analyze the dynamics of the Turkish Treasury bond yield curve over time and to evaluate the performance of various forecasting models. The spot yield data used in the study were obtained from the Bloomberg Terminal via the Bloomberg Valuation Service (BVAL) code. When valuing fixed income securities in Turkey, Bloomberg takes into account data from the Central Bank of the Republic of Turkey (CBRT), price quotes from market-making institutions, and bond yields reported by local banks. In addition, it performs cross-validation using price movements from Borsa Istanbul (BIST) and international bond markets to enhance data accuracy. In bond markets, not all maturities may be actively traded, and liquidity may be limited for certain tenors. In such cases, BVAL employs mathematical modeling and interpolation techniques to estimate bond yields for missing maturities. This process enables the construction of a reliable and up-to-date spot yield curve for Turkish Treasury bonds.

The following figure presents a three-dimensional visualization of the historical development of spot yields over the 2012–2025 period for the selected set of maturities.

Figure 4.1: *Yield Curve Surface*



The historical evolution of different yield series by maturity is illustrated in the graph below. It can be observed that yields have increased significantly in the recent period due to the sharp rise in inflation. During periods of rising yields, the divergence between maturities has also become more pronounced. Nevertheless, the yield curves tend to move largely in tandem, and since 2018, yields have generally remained at or above the 15% range. In contrast, during the period between 2012 and 2018, when inflation was relatively low, yields hovered around the 10% level.

Table 4.1 presents the average yield, minimum and maximum values, standard deviation, and autocorrelation coefficients at various lags for each maturity. Short-term bonds exhibit higher average yield levels compared to long-term bonds. For instance, the average yield of bonds with a 3-month maturity is 17.13%, while for 10-year maturity bonds it is 14.58%. Contrary to typical market conditions, average yields tend to decline as maturities increase. This phenomenon is attributed to recent exchange rate and inflation shocks in the Turkish economy.

Moreover, short-term bonds display higher levels of volatility. For example, the standard deviation of the 3-month bond yield is 10.29, whereas it is 5.57 for the 10-year bond. In other words, short-term bonds tend to offer higher average returns but are also associated with greater risk and volatility.

On average, the yield curve exhibits a negative slope, indicating an inverted yield structure. Additionally, the high autocorrelation coefficients confirm the strong time dependency of yield movements.

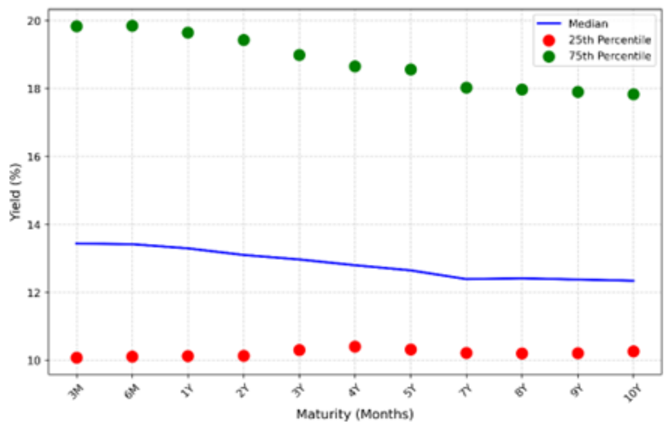
Table 4.1: *Descriptive Statistics*

Maturity	**Mean**	**Min**	**Max**	**Std**	ρ_{1M}	ρ_{12M}	ρ_{30M}
3M	17.13	5.58	49.12	10.28	0.99	0.98	0.96
6M	17.03	5.60	47.57	10.02	0.99	0.98	0.96
1Y	16.79	5.66	44.44	9.40	0.99	0.98	0.96
2Y	16.26	5.61	38.84	8.34	0.99	0.98	0.95
3Y	15.86	5.70	34.70	7.48	0.99	0.98	0.94
4Y	15.50	5.84	32.82	6.89	0.99	0.97	0.94
5Y	15.20	6.02	31.56	6.49	0.99	0.97	0.94
7Y	14.75	6.35	29.44	5.91	0.99	0.97	0.94
8Y	14.65	6.48	28.65	5.71	0.99	0.97	0.94
9Y	14.56	6.59	28.13	5.57	0.99	0.97	0.94
10Y	14.58	6.70	27.99	5.57	0.99	0.97	0.94
Slope	-2.55	-4.62	6.25	5.79	0.99	0.97	0.91
Curvature	0.82	-3.56	7.94	1.67	0.98	0.82	0.61

To visualize the central tendency and volatility structure of bond yields across different maturities presented in Figure 4.3, a distribution plot was constructed using the median values along with the 25th and 75th percentiles of the yield data. The median line generally exhibits a stable and slightly downward-sloping trend. It is also visually apparent that short-term bonds have both higher median yields and a wider Interquartile

Range (IQR) , indicating greater dispersion within the 25th to 75th percentile interval.

Figure 4.2: *Yield Curve Distribution*

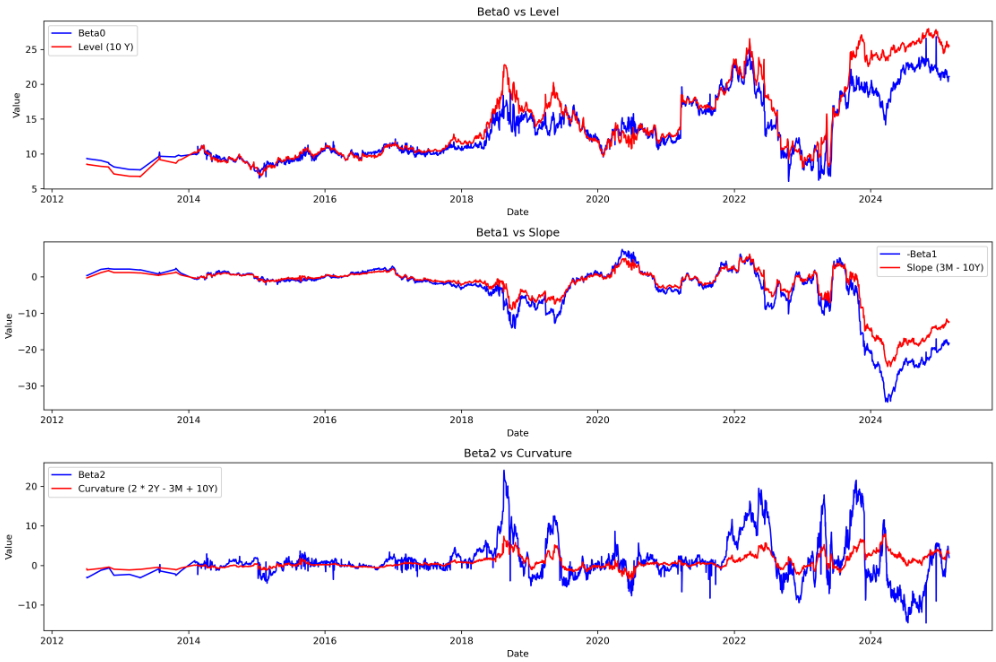


The three-factor model proposed by Nelson and Siegel (1987) plays a significant role in capturing the shape of the yield curve [36].In this model, the betas (latent factors) represent key parameters that help to explain the evolution of interest rates over time. These factors are commonly interpreted as level, slope, and curvature. The parameters must be estimated using nonlinear models; however, most approaches in the literature assume a fixed value for the decay parameter τ, which allows the betas to be estimated using the Ordinary Least Squares (OLS) method. τ determines the maturity at which the medium-term (curvature) factor reaches its peak. In practice, a commonly accepted choice for τ is 30 months, which has been shown to provide effective estimates of the level, slope, and curvature components.

The OLS method enables the estimation of betas for each month, allowing the yield curve to take on different shapes over time. The beta estimates obtained via OLS reflect the time-varying behavior of the level, slope, and curvature of the yield curve.

In Figure 4.4, the latent factors are compared based on both model-derived and data-based approaches. The β_0 parameter, which represents the level, is compared with the observed 10-year bond yield. The slope is calculated as the difference between the 10-year and 3-month yields, while the curvature is defined as twice the 2-year yield minus the sum of the 3-month and 10-year yields. The graphical analysis indicates that the model provides a good fit to the observed data.

Figure 4.3: *Model-Derived vs. Data-Based Beta Parameters*



Descriptive statistics and autocorrelation analysis results for each parameter are presented in the table below. The findings indicate that the parameters β_0, β_1, and β_2 are generally highly dependent on each other over time. However, this dependency appears to weaken for β_2 over shorter time horizons. The strong correlation between β_0 and β_1 in both short- and medium-term lags suggests that these parameters exhibit relatively stable and consistent behavior over time.

In contrast, β_2 shows greater fluctuations and a notable decline in long-term autocorrelation, indicating that this parameter behaves in a less consistent and more volatile manner. This difference highlights the dynamic nature of the curvature component compared to the more stable level and slope factors.

Table 4.2: *Descriptive statistics and autocorrelation coefficients for latent factors.*

Parameter	**Mean**	**Min**	**Max**	**Std**	ρ_{1M}	ρ_{12M}	ρ_{30M}
β_0	13.38	4.40	25.72	4.96	0.98	0.93	0.87
β_1	3.97	-7.71	34.75	7.79	0.99	0.95	0.88
β_2	0.44	-23.48	26.88	6.84	0.97	0.81	0.58

4.2 Model Specifications and Methodological Design

This section provides a detailed explanation of the models used in the study, including their definitions, core components, and operational mechanisms. Understanding the structure of the models offers a fundamental basis for interpreting the methods and parameters employed in the analysis. Within this context, the definitions and structures of key models such as the Nelson-Siegel Yield Curve Model and various Time Series Models are examined.

The primary objective of the study is to evaluate the effectiveness of deep learning models in forecasting the yield curve. Accordingly, methods and models commonly used in the yield curve forecasting literature have been implemented within a deep learning framework, and the results have been compared with those obtained from classical econometric approaches and naïve models.

The models utilized in this study can be grouped into two main categories. The first category includes models that directly forecast yield rates without relying on any specific curve-fitting or parametric model, such as the Nelson-Siegel framework. The second category involves models that incorporate yields derived from the Nelson-Siegel model as part of the forecasting process.

At the conclusion of the study, the results of all models are evaluated from a holistic perspective to assess their comparative performance and forecasting accuracy.

4.2.1 Long Short-Term Memory (LSTM Model)

In this study, the LSTM model—known for its ability to learn long-term dependencies in time series data—was tested in different architectures to forecast the yield curve. Initially, yield rates were directly fed into the LSTM model as inputs, and forward-looking forecasts were generated. This setup represents a basic LSTM application, allowing for the prediction of future yields based solely on historical data.

Subsequently, the model structure was modified by incorporating a Nelson-Siegel layer, and a new architecture was designed based on forecasts produced through this layer. This enhanced model was developed under two different configurations: one with a fixed tau parameter and another allowing for a dynamic tau. While the fixed tau model maintains a constant tau parameter within the Nelson-Siegel framework, the dynamic tau model allows this parameter to vary during training. Both configurations aim to provide a more flexible and adaptive representation of the yield curve.

In the final configuration, yield forecasts were generated through an LSTM model trained on latent factors extracted from the Nelson-Siegel model—namely, level, slope, and curvature. This approach enables a deeper modeling of the yield curve structure and aims to improve forecasting accuracy by leveraging these underlying components.

To apply the LSTM model effectively, the data must be appropriately preprocessed and structured. Using a windowing technique, the time series data were segmented into input and target pairs based on a defined window size (e.g., n = 1). Target values were aligned with future dates, taking holidays and non-trading days into account, and rows with missing or invalid dates were removed.

In the standard LSTM model, forecasts were generated using a deep learning-based architecture both directly from yield data and from latent factors estimated via OLS. In the version trained directly on yields, the input to the model was an 11-dimensional vector, representing yields across different maturities. The model, trained on these 11 features, simultaneously predicted 11 yield values as output.

The model architecture—including hidden layers, the number of nodes, and related hyperparameters—was optimized using various search strategies. The configuration with the best in-sample fit was selected. Similarly, other key hyperparameters such as batch size, number of epochs, and the optimization algorithm were determined through grid search and Bayesian optimization methods, with consideration for commonly used values in the literature. While there is no universally accepted method for hyperparameter selection in the literature, this study sought to identify optimal values in a data-driven and objective manner. The hyperparameters and their corresponding values used in the final model are presented in the table below.

Table 4.3: *Hyperparameter comparison across different LSTM-based models.*

Hyperparameter	**LSTM (on Yields)**	**DLNS (fix tau)**	**DLNS**	**LSTM (on Latents)**
LSTM Units	78	180	78	64
Dropout	0.1903	0.3127	0.1903	0.2
Dense Units	102	44	102	64
L2 Regularization	0.002%	0.001%	0.002%	0.10%
Learning Rate	0.026%	0.067%	0.026%	0.03%
Batch Size	64	64	64	64
Epochs	500	500	500	250
Early Stopping Patience	100	100	100	100
Optimizer	Adamprop	Adamprop	Adamprop	Adamprop
Activation Functions	ReLU	ReLU	ReLU	ReLU
Cost	MSE	MSE	MSE	MSE

In order to enhance the predictability of interest rate term structures and better analyze the dynamics of the yield curve, the Deep Learning Nelson-Siegel (DLNS) model was developed (Sang-Heon Lee,2023). The model incorporates a Nelson-Siegel (NS) layer that takes latent factors derived from a deep learning model as input and transforms them into a smooth and continuous yield curve. This NS layer acts as an integrated component that optimizes yield curve predictions across multiple maturities and serves as the model's output layer.

The output of the NS layer consists of term structure forecasts directly derived from the model's latent factors, resulting in yield curves that are both continuous and differentiable. In the DLNS model, the latent factors (β_1, β_2, β_3, and τ) are examined under two different configurations: one with fixed τ and the other with dynamic τ. The LSTM model's ability to capture complex nonlinear dependencies provides a powerful framework for estimating these latent factors.

In this regard, the DLNS model offers a flexible architecture that allows the Nelson-Siegel τ parameter (also denoted as λ) to vary dynamically. Given that there is no universally accepted method for determining the optimal value of τ in the literature, this dynamic approach not only addresses a significant gap but also highlights the model's adaptability.

By allowing parameters to vary over time, the DLNS model is capable of estimating the term structure of interest rates as a continuous curve. This approach overcomes the limitations of traditional econometric models that treat each maturity yield as an independent forecast. The DLNS framework aims to generate more accurate and consistent forecasts by directly modeling the relationships between different maturities and capturing the underlying latent factors—level, slope, and curvature—of the yield curve.

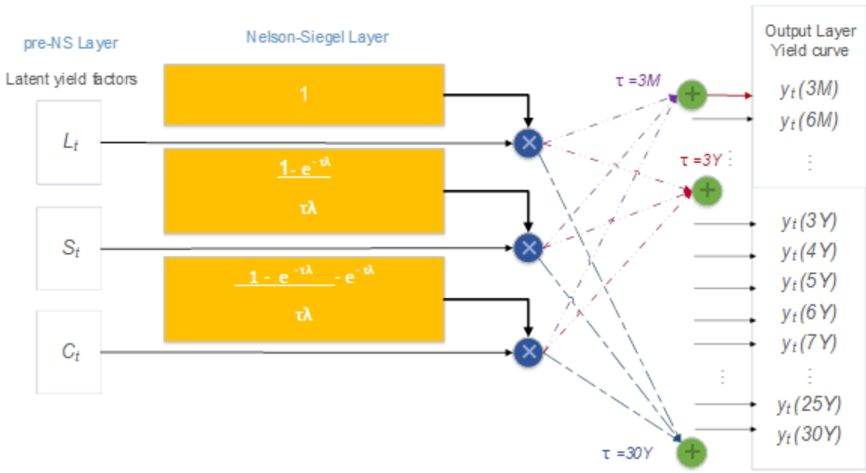
4.2.1.1 DLNS Model with Fixed λ

The Deep Learning Nelson-Siegel (DLNS) model is an extension of traditional yield curve modeling that integrates deep learning methodologies. By incorporating a

Nelson-Siegel (NS) layer into the deep learning architecture, the DLNS model enables the joint estimation of both the yield factors and the corresponding latent variables, ultimately producing a smooth and interpretable yield curve forecast.

A critical component of the model is the time decay parameter (λ), which governs the weighting of different maturities in the Nelson-Siegel specification. The DLNS model can be implemented in two primary forms: one with a fixed λ and the other with a time-varying λ.

Figure 4.4: *DLNS Model with Fixed λ*



In the fixed-λ formulation, the NS layer is embedded in the deep learning architecture as a predefined functional component. This layer mimics the factor loading structure of the standard Nelson-Siegel model but is implemented without trainable parameters, acting as a static transformation within the network.

The output of the encoder block—referred to as the pre-NS layer—serves as the input to the NS layer, where it is transformed into a yield curve through the use of fixed Nelson-Siegel factor loadings. Since λ remains constant throughout training, the NS layer does not introduce additional degrees of freedom, allowing for a well-defined and stable backpropagation process, where the gradient vector remains continuous and differentiable.

The fixed-λ DLNS model can be mathematically represented as follows:

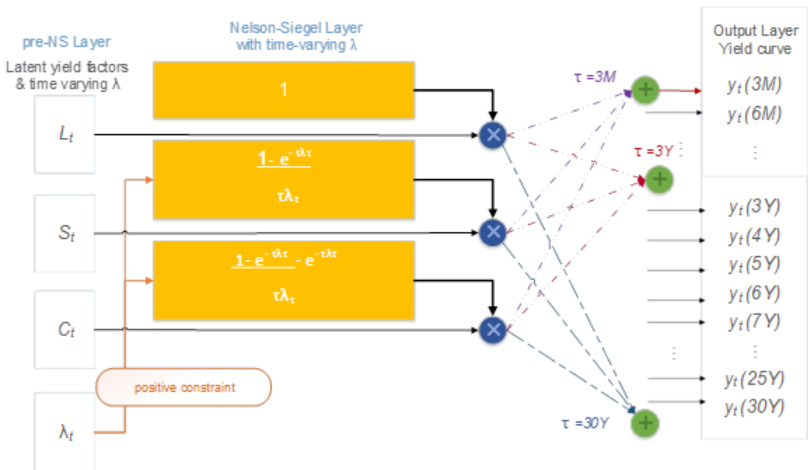
$$\mathrm{NS}(f) = \mathrm{DL}(y) \tag{4.1}$$

where $\mathrm{DL}(y)$ represents the deep learning component that extracts latent yield factors, and $\mathrm{NS}(f)$ denotes the Nelson-Siegel transformation applied to these factors.

4.2.1.2 DLNS Model with Time-Varying λ

While the fixed-λ formulation provides a stable foundation for yield curve modeling, it does not account for variations in the time decay parameter under different market conditions. To overcome this limitation, the DLNS model can be extended to incorporate a time-varying λ, treating it as an additional latent factor.

Figure 4.5: *DLNS Model with time-varying λ*



In this formulation, λ is no longer a fixed hyperparameter, but rather a dynamic quantity that evolves over time. λ must be transformed into positive values. This transformation is implemented using a Sigmoid function to map unconstrained values into a predefined range, ensuring that the curvature factor loading is maximized within an appropriate time frame. This step is essential to guarantee that the curvature component is most effective within realistic maturity horizons.

$$\lambda = \text{Sigmoid}(\lambda) \times \left(\frac{1.8}{24} - \frac{1.8}{60}\right) + \frac{1.8}{60} \tag{4.2}$$

This transformation ensures that the predicted λ values stay within a reasonable range and maintains the interpretability of the Nelson-Siegel factors.

The internal structure of the NS layer allows for a well-defined backpropagation process even with a time-varying λ. The gradient vector of the yield curve output with respect to the latent yield factors remains continuous and can be computed as follows:

$$\frac{\partial y(\tau)}{\partial f_t} = \begin{bmatrix} \frac{\partial y(\tau)}{\partial \beta_{0,t}} \\ \frac{\partial y(\tau)}{\partial \beta_{1,t}} \\ \frac{\partial y(\tau)}{\partial \beta_{2,t}} \\ \frac{\partial y(\tau)}{\partial \lambda_t} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau} \\ \frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau} \\ \frac{\beta_2}{\lambda_t}(\lambda_t \tau e^{-\lambda_t \tau}) - \frac{(\beta_1+\beta_2)}{\lambda_t}\left(\frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau}\right) \end{bmatrix} \tag{4.3}$$

Here, f_t represents the vector of latent yield factors: $(\beta_0, \beta_1, \beta_2, \lambda)$.

Within this framework, by incorporating a time-varying λ, the DLNS model is designed to adapt more effectively to changing market conditions. Another advantage of the model is that it provides a solution to estimating τ, a parameter that is typically difficult to obtain and forecast. By optimizing the model, it becomes possible to examine the latent factors it identifies.

In this respect, the model not only performs yield forecasting but also produces β parameters and the τ value as outputs. This allows for deeper interpretation of the results and provides valuable insights into the dynamics of the yield curve.

4.2.1.3 LSTM Based on Beta Factors

By leveraging the Long Short-Term Memory (LSTM) model's ability to capture long-term dependencies in time series data, the aim is to forecast the future evolution of Nelson-Siegel factors and thereby predict the yield curve. In this context, a separate LSTM model is constructed for each latent factor to forecast its future values. Using the obtained forward-looking latent factors, yield forecasts are generated.

The input values for the model, namely the Nelson-Siegel betas, are estimated using the Ordinary Least Squares (OLS) method with a fixed τ value. These latent factors are then used to train the LSTM model, enabling forward-looking predictions. Based on the predicted latent factors, the yield curve is reconstructed and estimated yield rates for different maturities are obtained.

$$\hat{\beta}_{i,t+h} = f_{\text{LSTM}}\left(\beta_{i,t}, \beta_{i,t-1}, \ldots, \beta_{i,t-k}\right) \tag{4.4}$$

Here, $\hat{\beta}_{i,t+h}$ denotes the predicted value of the i-th latent factor h steps ahead, f_{LSTM} represents the function learned by the LSTM model, $\beta_{i,t}$ is the current value of the i-th latent factor at time t, and k denotes the length of the time window formed by past data points used in the model.

4.2.1.4 Autoregressive (AR) and Random Walk (RW) Models

To forecast the future evolution of the Nelson-Siegel factors, in addition to the LSTM model, the Autoregressive (AR) model and the Random Walk (RW) model have also been applied. The AR model is a time series model used to predict future values based on past observations. In this study, a separate AR model is constructed for each latent factor (β_0, β_1, β_2) using the following formulation:

$$\hat{\beta}_{i,t+h} = \alpha + \phi\beta_{i,t} + \varepsilon_t \tag{4.5}$$

Here, $\hat{\beta}_{i,t+h}$ represents the forecasted value of the i-th latent factor h steps ahead, $\beta_{i,t}$ is the observed latent factor at time t, α and ϕ are the coefficients learned by the model, and ε_t denotes the error term. This model is trained using Ordinary Least Squares (OLS) regression and applied individually to each latent factor.

The AR model is also applied directly to the yield values instead of the Nelson-Siegel factors, and the results are compared. The aim here is to evaluate the model's performance relative to the latent factor-based approach by forecasting future yields using the historical values of yields at different maturities:

$$\hat{y}_{t+h} = \alpha + \phi y_t + \varepsilon_t \tag{4.6}$$

In this equation, $\hat{y}_{t+h}$ is the predicted yield h steps ahead, y_t is the current yield at time t, and α, ϕ, and ε_t retain their respective definitions.

The final model used in this study to forecast the evolution of yields over time is the Random Walk (RW) model. The RW model is based on the assumption that the future values of a time series will be similar to their current values. As such, the RW model does not involve any training or coefficient estimation between the current and predicted yields. It does not learn trends or seasonality but simply assumes that current values will persist in the future. Despite its simplicity, the RW model is considered robust and widely used in the financial forecasting literature.

$$\hat{y}_{t+h} = y_t + \varepsilon_t \tag{4.7}$$

Here, $\hat{y}_{t+h}$ represents the predicted yield h steps ahead, y_t is the yield at time t, and ε_t is the error term.

The results obtained from these models will be evaluated in the following sections and compared with more complex models such as LSTM.

4.3 Forecast Results and Performance Comparison

To enhance the performance of the LSTM models used for yield forecasting and to determine appropriate hyperparameter calibration, Bayesian Optimization and Grid Search methods were employed for hyperparameter tuning. The models used in this study are designed to forecast yield curve rates and start with an LSTM layer that takes 11 input features. The model architecture includes a Dropout layer, Dense (fully connected) layers, and a custom Lambda layer that incorporates the Nelson-Siegel function. The output layer produces either three parameters when tau is fixed, or four parameters when tau is dynamic.

Two different methods were used for hyperparameter optimization: Bayesian Optimization and Grid Search. Bayesian Optimization provides a more efficient and targeted hyperparameter search by intelligently narrowing the search space. With this method, hyperparameters such as the number of LSTM units, dropout rate, learning rate, and L2 regularization coefficient were optimized.

Grid Search, on the other hand, is a systematic method that evaluates all possible combinations to identify the best-performing setup. In this study, activation functions (ReLu, tanh, sigmoid) and batch sizes (32 and 64) were optimized using Grid Search. For each combination, the model was trained, and the validation loss was computed to identify the optimal activation function and batch size. By using these two different optimization strategies, model prediction accuracy was improved, and the risk of overfitting was minimized. While Bayesian Optimization is more flexible in optimizing continuous hyperparameters, Grid Search is particularly useful for systematically testing discrete values. This approach effectively combines the flexibility of Bayesian Optimization with the structured nature of Grid Search to achieve the best possible model configuration.

Throughout the training process of the LSTM model with time-varying tau, the loss functions and error metrics were monitored. It was observed that both training and validation loss decreased consistently. Moreover, the downward trend in Mean Absolute Error (MAE) for both training and validation sets demonstrates that the model not only fits the training data well but also generalizes effectively to unseen data. The results indicate that the model continues to learn over time, with no significant signs of overfitting or underfitting. The fact that training and validation losses decrease at similar levels suggests that the model performs in a balanced manner across different datasets.

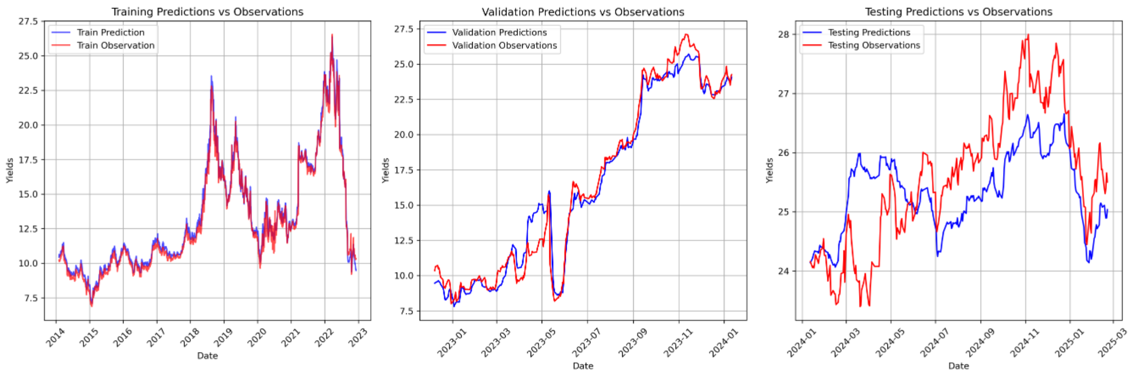
Figure 4.6: *Model Loss and MAE During Training*

To evaluate the model's prediction performance on the training, validation, and test sets, the predicted values were compared with the actual observed values. The plots show that the model's predictions closely follow the observed values, indicating that the model was able to successfully apply the patterns learned from the training data to the validation and test datasets.

To assess the predictive accuracy of the model, the RMSE metric was examined and found to be at a satisfactory level. Furthermore, the absence of a significant error gap between the training and validation sets indicates that the model does not suffer from overfitting and demonstrates strong generalization capability.

One of the most significant advantages of this model is its ability to estimate latent

Figure 4.7: *Model Loss and MAE During Training*



factors and the λ parameter without requiring any prior assumptions or predefined constraints. The resulting β parameters and τ value are presented in Figure **??**. It is observed that the latent factors optimized by the model through the backpropagation algorithm are fitted remarkably well.

Moreover, the model successfully captures the time-varying dynamics of λ (τ), a parameter for which no definitive method has been established in the literature for its dynamic determination over time. These results demonstrate that the model can serve as a powerful forecasting tool within the Nelson-Siegel framework.

Similarly, the other models were trained within the same framework, and their prediction performances were evaluated. During the training process of each model, similar dataset structures and optimization strategies were applied. In the following section of the study, the performance of the LSTM, AR, and RW models will be assessed using an out-of-sample analysis.

Figure 4.8: *Comparison of Estimated and Parametrically Calculated Latent Factors*

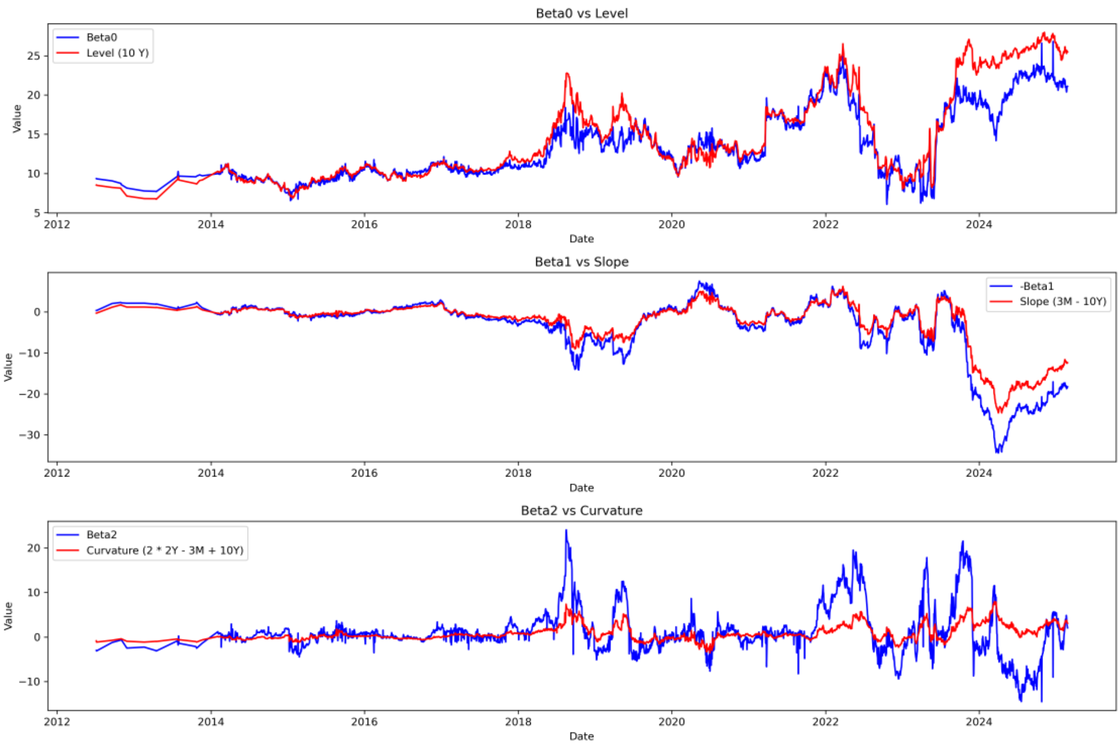
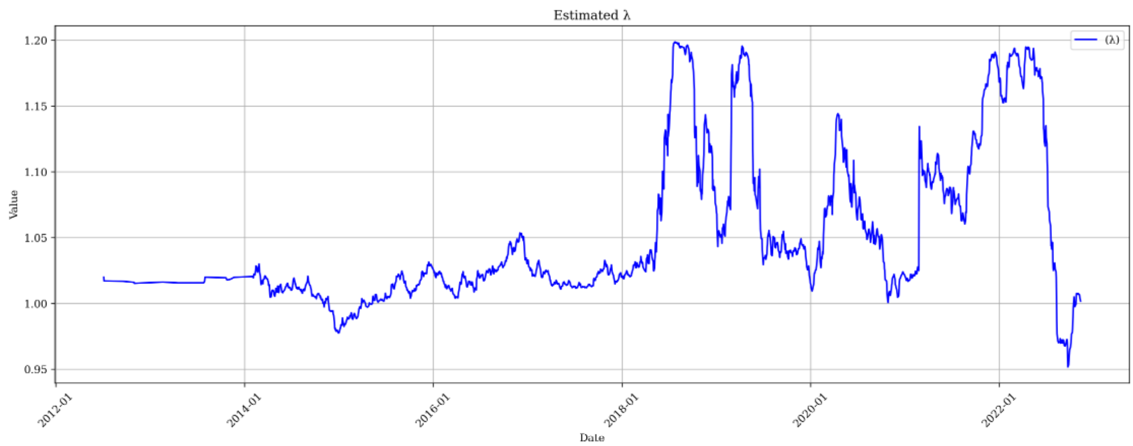


Figure 4.9: *Estimated* λ



4.3.1 Out-of-Sample Performance Evaluation

To evaluate the predictive power of the models, the Out-of-Sample (OOS) forecasting method was employed. Since LSTM models tend to perform better with larger training datasets, the expanding rolling window approach was used to ensure a more accurate assessment of their forecasting ability.

Forecasts were generated for different horizons: 1, 30, 90, 120, and 360 days ahead. The predicted yield values obtained from the models were compared with the actual observed yield values. To assess model performance, the Root Mean Squared Error (RMSE) was used as the primary evaluation metric. Furthermore, to test whether there is a statistically significant difference in the forecast accuracy between two competing models, the Diebold-Mariano (DM) test was applied.

$$\text{RMSE} = \sqrt{\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(y_t(\tau_i) - \hat{y}_t(\tau_i)\right)^2} \tag{4.8}$$

To evaluate the predictive performance of the models, forecasts were generated for multiple horizons using an out-of-sample framework. The forecasting results, summarized in Table 4.4, highlight each model's performance across different horizons.

Table 4.4: *OOS Results*

Models	**1 Day**	**30 Day**	**90 Day**	**120 Day**	**360 Day**
DLNS Model (Fixed λ)	1.89	3.79	4.89	4.98	6.75
DLNS Model (Time-varying λ)	0.73	2.35	3.07	3.59	4.85
AR Model (on yields)	0.22	2.39	5.33	6.10	6.65
RW Model (on yields)	0.21	2.07	4.14	4.76	7.04
LSTM Model (on latents)	0.69	2.09	2.56	2.93	4.83
AR Model (on latents)	0.56	2.04	3.72	3.84	5.70
LSTM Model (on yields)	1.58	3.69	4.69	4.61	6.55

The analysis reveals that simpler models, such as the AR model trained on observed yields and the RW model, perform exceptionally well in short-term forecasts, producing the lowest RMSE values among all models. However, their performance declines noticeably as the forecast horizon extends, particularly beyond 90 days.

In contrast, deep learning-based models demonstrate stronger generalization capabilities over medium- and long-term horizons. Notably, the Deep Learning Nelson-Siegel (DLNS) model with time-varying λ and the LSTM model trained on latent factors consistently achieve the lowest RMSE values at 90-, 120-, and 360-day horizons. These results suggest that models incorporating latent structure and dynamic factor modeling offer enhanced forecasting power, especially when predicting longer-term yield curve dynamics.

On the other hand, models such as the DLNS with fixed λ and the LSTM model without the Nelson-Siegel structure (i.e., the yield-based LSTM model) show relatively weaker performance across all horizons, highlighting the importance of structured factor modeling in improving yield predictions.

To further assess the statistical significance of forecast accuracy differences, the Diebold-Mariano (DM) test was applied. The DM test results presented in Table 4.5 provide a comprehensive comparison of forecasting performance across various models and forecast horizons. In this analysis, the RW model was used as the benchmark model, and all other models were statistically compared against it.

In 1-day short-term forecasts, both the AR model (trained on yields) and the LSTM model (trained on latent factors) produced positive and statistically significant DM values, indicating superior performance relative to the benchmark RW model. While the RW model recorded the lowest RMSE value at the 1-day horizon, this apparent contradiction can be explained by the different nature of the two evaluation metrics. RMSE captures the absolute magnitude of forecast errors, whereas the DM test evaluates whether

Table 4.5: *DM Test Results*

Models	1 Day	30 Days	90 Days	120 Days	360 Days
DLNS Model (Fixed λ)	-3.983^{**}	-2.935^{*}	-0.383^{*}	1.750^{*}	6.824^{*}
DLNS Model (Time-varying λ)	-2.794^{*}	-5.415^{*}	-5.314^{*}	-4.602^{*}	3.796^{*}
AR Model (on yields)	3.737^{**}	3.108^{*}	3.865^{*}	3.941^{*}	3.528^{*}
LSTM Model (on latents)	6.324^{**}	4.640^{*}	4.840^{**}	5.340^{**}	8.738^{*}
AR Model (on latents)	-2.530^{*}	3.313^{*}	6.441^{**}	5.698^{**}	4.419^{**}
LSTM Model (on yields)	-4.100^{**}	-2.188^{*}	-0.661^{*}	1.890^{*}	6.153^{*}

the difference in forecast errors between two models is statistically significant and consistent over time. A model may exhibit a lower RMSE due to a few exceptionally accurate predictions, but if its performance is less stable, the DM test may not favor it. Conversely, a model with slightly higher RMSE but more consistent accuracy may be judged statistically superior. Therefore, the positive and significant DM statistics for both the AR and LSTM models suggest that, despite RW's slightly lower average error, these models offered more reliable and systematically better forecasts at the 1-day horizon.

At the 30-day medium-term forecast horizon, several models produced positive and statistically significant DM statistics, indicating a clear improvement in forecasting accuracy compared to the benchmark RW model. In particular, the LSTM model trained on latent factors and the AR model trained on latent factors produced successful results in this period. However, DLNS models continued to perform worse than the benchmark, as indicated by their negative and significant DM values. This may point to certain adaptation issues in models with time-varying structures during this horizon.

For the 90- and 120-day forecast horizons, the AR model trained on yields, as well as the AR and LSTM models trained on latent factors, stood out with positive and significant DM statistics. These results demonstrate that, in the medium term, both traditional and deep learning-based models can yield successful forecasts when properly structured and calibrated.

In the 360-day long-term forecasts, the overall picture changes significantly. The LSTM model trained on latent factors and DLNS models showed strong and statistically significant positive DM values, indicating a clear advantage over the benchmark model. This suggests that deep learning-based models can gradually learn and capture complex patterns, becoming more effective in long-term forecasting. It is evident that models which struggle in the short term may eventually outperform the benchmark over longer forecast horizons.

5. CONCLUSIONS

This thesis has investigated the modeling and forecasting of the Turkish Treasury bond yield curve using both traditional econometric methods and modern deep learning approaches. The study implemented models such as AR, RW, the Dynamic Nelson-Siegel (DLNS) model, and Long Short-Term Memory (LSTM) neural networks to assess their forecasting accuracy over different horizons. Forecast evaluations were conducted using an expanding window out-of-sample framework, with RMSE as the primary error metric and the Diebold-Mariano test for statistical significance comparisons.

The empirical results reveal several important insights regarding model performance across different forecast horizons. In short-term forecasts (1-day horizon), simpler models such as the RW and AR models perform well in terms of RMSE. However, the Diebold-Mariano test indicates that both the AR model based on observed yields and the LSTM model trained on latent factors produce statistically superior and more consistent forecasts compared to the RW benchmark, suggesting a stronger generalization capacity even at very short horizons.

As the forecast horizon extends to the medium term (30 to 120 days), the advantage of deep learning approaches becomes more pronounced. In particular, LSTM and AR models trained on latent factors consistently outperform the benchmark model in both accuracy and statistical significance. These results suggest that models utilizing latent structure are more capable of capturing the underlying dynamics of the yield curve than naïve alternatives.

For long-term forecasts (360 days), the performance gap between traditional models and deep learning models becomes even more evident. The LSTM model trained on latent factors and the DLNS model with a time-varying decay parameter (λ) exhibit superior predictive accuracy, highlighting the ability of deep learning models to model complex, nonlinear, and persistent patterns in the evolution of the yield curve.

The findings of the Diebold-Mariano tests further underline the role of model complexity and structure in forecasting performance. While simple models may offer advantages in the short term, more flexible and structured models such as LSTM and DLNS exhibit growing advantages as the forecast horizon extends. These results emphasize that no single model universally dominates across all horizons, and the model choice must be aligned with the specific forecasting goal and time frame.

Ultimately, this study demonstrates that deep learning models can significantly enhance forecasting accuracy, especially for medium- and long-term horizons. These models offer valuable tools for both investors and policymakers in managing risk, formulating expectations, and making informed decisions in the fixed-income market.

Future research may focus on hybrid modeling approaches that combine deep learning methods with macroeconomic indicators, or explore attention-based architectures such as Transformers for further improvements in sequential modeling. Additionally, real-time forecasting under regime changes and stress conditions presents a valuable direction for robustness testing and policy analysis.

REFERENCES

[1] BIS, "Zero-coupon yield curves: technical documentation," BIS Papers 25, Bank for International Settlements, 2005.

[2] P. Nymand-Andersen, "Yield curve modelling and a conceptual framework for estimating yield curves: evidence from the european central bank's yield curve," tech. rep., European Central Bank, 2018. ECB Document.

[3] D. J. Bolder and S. Gusba, "Exponentials, polynomials, and fourier series: More yield curve modelling at the bank of canada," tech. rep., Bank of Canada, 2002.

[4] N. Anderson, "Estimating and interpreting the yield curve," 1996. No Title.

[5] O. Vasicek, "An equilibrium characterization of the term structure," *Journal of Financial Economics*, vol. 5, no. 2, pp. 177–188, 1977.

[6] J. C. Cox, J. E. Ingersoll, and S. A. Ross, "A theory of the term structure of interest rates," *Econometrica*, vol. 53, no. 2, pp. 385–407, 1985.

[7] L. U. Dothan, "On the term structure of interest rates," *Journal of Financial Economics*, vol. 6, no. 1, pp. 59–69, 1978.

[8] F. A. Longstaff and E. S. Schwartz, "Interest rate volatility and the term structure: A two-factor general equilibrium model," *The Journal of Finance*, vol. 47, no. 4, pp. 1259–1282, 1992.

[9] T. S. Ho and S. B. Lee, "Term structure movements and pricing interest rate contingent claims," *The Journal of Finance*, vol. 41, no. 5, pp. 1011–1029, 1986.

[10] J. Hull and A. White, "The use of the control variate technique in option pricing," *Journal of Financial and Quantitative Analysis*, vol. 23, no. 3, pp. 237–251, 1988.

[11] F. Black, E. Derman, and W. Toy, "A one-factor model of interest rates and its application to treasury bond options," *Financial Analysts Journal*, vol. 46, no. 1, pp. 33–39, 1990.

[12] D. Heath, R. Jarrow, A. Morton, and M. Spindel, "Easier done than said," *Risk*, vol. 5, no. 9, pp. 77–80, 1992.

[13] J. H. McCulloch, "Measuring the term structure of interest rates," *The Journal of Business*, vol. 44, no. 1, pp. 19–31, 1971.

[14] O. A. Vasicek and H. Fong, "Term structure modeling using exponential splines," *The Journal of Finance*, vol. 37, no. 2, pp. 339–348, 1982.

[15] M. Fisher, D. W. Nychka, and D. Zervos, “Fitting the term structure of interest rates with smoothing splines,” tech. rep., 2000. Working Paper or Preprint.

[16] R. B. Litterman, J. Scheinkman, and L. Weiss, “Volatility and the yield curve,” *The Journal of Fixed Income*, vol. 1, no. 1, pp. 49–53, 1991.

[17] R. S. G”urkaynak, B. Sack, and J. H. Wright, “The us treasury yield curve: 1961 to the present,” *Journal of Monetary Economics*, vol. 54, no. 8, pp. 2291–2304, 2007.

[18] L. E. Svensson, “Estimating and interpreting forward interest rates: Sweden 1992-1994,” tech. rep., 1994. Working Paper or Preprint.

[19] T. Bj”ork and B. J. Christensen, “Interest rate dynamics and consistent forward rate curves,” *Mathematical Finance*, vol. 9, no. 4, pp. 323–348, 1999.

[20] F. X. Diebold, G. D. Rudebusch, and S. B. Aruoba, “The macroeconomy and the yield curve: a dynamic latent factor approach,” *Journal of Econometrics*, vol. 131, no. 1-2, pp. 309–338, 2006.

[21] M. De Pooter, “Examining the nelson-siegel class of term structure models: In-sample fit versus out-of-sample forecasting performance,” 2007. Available at SSRN 992748.

[22] F. X. Diebold, M. Piazzesi, and G. D. Rudebusch, “Modeling bond yields in finance and macroeconomics,” *American Economic Review*, vol. 95, no. 2, pp. 415–420, 2005.

[23] F. X. Diebold and C. Li, “Forecasting the term structure of government bond yields,” *Journal of Econometrics*, vol. 130, no. 2, pp. 337–364, 2006.

[24] M. Fernandes and F. Vieira, “A dynamic nelson–siegel model with forward-looking macroeconomic factors for the yield curve in the us,” *Journal of Economic Dynamics and Control*, vol. 106, p. 103720, 2019.

[25] J. H. Christensen, F. X. Diebold, and G. D. Rudebusch, “An arbitrage-free generalized nelson–siegel term structure model,” *Econometrics Journal*, 2009. In press or working paper.

[26] K. Nyholm, “A rotated dynamic nelson-siegel model with macro-financial applications,” Tech. Rep. 1851, European Central Bank, 2015. ECB Working Paper.

[27] S. J. Koopman, M. I. Mallee, and M. van der Wel, “Analyzing the term structure of interest rates using the dynamic nelson–siegel model with time-varying parameters,” *Journal of Business & Economic Statistics*, vol. 28, no. 3, pp. 329–343, 2010.

[28] R. Gimeno and J. M. Nave, “A genetic algorithm estimation of the term structure of interest rates,” *Computational Statistics & Data Analysis*, vol. 53, no. 6, pp. 2236–2250, 2009.

[29] M. Gilli, S. Grosse, and E. Schumann, "Calibrating the nelson-siegel-svensson model," tech. rep., 2010. Available at SSRN 1676747.

[30] S.-H. Lee, "Second curvature factor and term premium in the us yield curve: Insights from deep learning," tech. rep., 2023. Available at SSRN 4652774.

[31] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020.

[32] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] C. R. Nelson and A. F. Siegel, "Parsimonious modeling of yield curves," *Journal of Business*, pp. 473–489, 1987.