



UNIVERSITÀ
DI PAVIA

Dipartimento di Scienze Economiche e Aziendali

Corso di Laurea Magistrale in Finance

**EUR/USD RETURN FORECASTING: A
COMPARATIVE ANALYSIS OF ECONOMETRIC,
NONLINEAR, MACHINE LEARNING AND DEEP
LEARNING MODELS**

Relatore:

Chiar.mo Prof. Paolo Giudici

**Tesi di Laurea
di Terni Valentina**

Matr. n. 551742

Anno Accademico 2025-2026

INDEX

1	EMPIRICAL ANALYSIS	3
1.1	CASE STUDY	3
1.2	DATA DESCRIPTION	4
1.3	DATASET PREPARATION	6
1.4	AUTOCORRELATION AND PARTIAL AUTOCORRELATION ANALYSIS	11
2	LINEAR REGRESSION MODELS.....	13
2.1	MULTIPLE LINEAR REGRESSION	14
2.1.1	<i>Residual Diagnostics</i>	17
2.1.2	<i>Stepwise Regression for OLS</i>	18
2.1.3	<i>HAC / Newey–West Robust Standard Errors</i>	20
2.2	RIDGE REGRESSION.....	22
2.3	LASSO REGRESSION	25
2.4	COMPARISON OF THE LINEAR MODELS.....	29
3	NON-LINEAR PARAMETRIC MODELS	31
3.1	POLYNOMIAL REGRESSION	31
3.2	SPLINE REGRESSION.....	33
4	MACHINE LEARNING MODELS	36
4.1	RANDOM FOREST.....	37
4.2	XGBOOST MODEL	40
4.3	SUPPORT VECTOR MACHINES WITH RBF KERNEL	44
5	DEEP LEARNING MODELS	48
5.1	MULTI-LAYER PERCEPTRON MODEL.....	50
5.2	LONG SHORT-TERM MEMORY MODEL	55
5.3	TEMPORAL FUSION TRANSFORMER MODEL	60
5.3.1	<i>Structure and features</i>	60
5.3.2	<i>Empirical Implementation</i>	62
5.3.3	<i>Variable Classification</i>	64
5.3.4	<i>Performance Plots</i>	65

5.3.6	<i>Interpretation and Feature Importance</i>	69
6	CONCLUSIONS	77
6.1	SUMMARY OF FINDINGS	77
6.2	IMPLICATIONS FOR EXCHANGE RATE FORECASTING	79
6.3	ECONOMIC IMPLICATIONS	80
6.4	LIMITATIONS AND FUTURE RESEARCH	82

1 EMPIRICAL ANALYSIS

1.1 Case Study

This empirical study focuses on forecasting the EUR/USD exchange rate and compares the performance of traditional econometric models with machine learning and deep learning approaches. The EUR/USD exchange rate was selected as the target variable because it is one of the most actively traded currency pairs in the foreign exchange market. Its high liquidity and the availability of extensive historical data make it particularly suitable for an empirical analysis of exchange rate forecasting. In addition, the EUR/USD exchange rate is affected by a wide range of economic and financial factors, including monetary policy, financial market conditions, macroeconomic developments, and investor expectations. These characteristics make it a useful case study for comparing different forecasting methods. Exchange rate movements can be difficult to predict because financial markets may exhibit nonlinear relationships, changes over time, and periods of increased volatility. Traditional linear models can provide useful and easily interpretable results, but they may not fully capture these features. For this reason, the analysis examines whether more flexible machine learning and deep learning models can achieve better forecasting results than traditional econometric approaches. The comparison also considers other aspects of the models, such as interpretability and their ability to capture relationships that may be difficult to represent through a linear specification. The dataset includes a range of explanatory variables from different areas of the financial market. In addition to other exchange rates, the dataset contains stock market indices, commodity prices, government bond yields, cryptocurrencies, and stablecoins. The inclusion of these variables is intended to provide a broader information set for forecasting the EUR/USD exchange rate. Considering different asset classes may also help identify relationships between the foreign exchange market and other segments of the financial system. The empirical analysis follows a chronological forecasting framework. The observations are ordered according to their dates so that the models use only information that would have been available at the time of each forecast. The dataset is divided into training, validation, and test sets. The training set is used to estimate the models, while the validation set is used for model selection and hyperparameter tuning. The final evaluation is then performed on the independent test set, which is not used

during the model development stage. The analysis starts with linear econometric models, which provide a benchmark and allow the relationships between the variables to be examined in a relatively simple and interpretable framework. More flexible nonlinear regression methods are then considered to account for possible nonlinear patterns in the data. The analysis subsequently moves to machine learning and deep learning models, including neural network architectures designed to work with sequential and time-series data. This structure allows the performance of models with different levels of complexity to be compared within the same forecasting framework. Model performance is evaluated using out-of-sample forecasts generated on the test set. Forecast accuracy is assessed using three commonly used regression metrics: the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), and the coefficient of determination (R^2). These measures provide different perspectives on model performance and allow the forecasting results obtained from the different approaches to be compared.

1.2 Data Description

The dataset used in this study consists of daily financial time series covering a five-year period, from March 2021 to March 2026. The data were obtained from publicly available financial databases, mainly “Yahoo Finance” and “Investing.com”, which provide reliable historical market information for financial research. The target variable of the analysis is the EUR/USD exchange rate, which represents the value of one euro in U.S. dollars. Due to its high trading volume and liquidity, EUR/USD provides a suitable case for analysing exchange rate dynamics and comparing different forecasting models. The dataset includes several explanatory variables selected to represent different segments of the financial market. These include other major exchange rates, equity market indices, commodity prices, government bond yields, cryptocurrencies, and stablecoins. The aim of including variables from different asset classes is to provide the models with a broader set of financial information that may be relevant for forecasting movements in the EUR/USD exchange rate. The explanatory variables included in the dataset are the following:

Name of the series	Symbol used	Description
EUR/USD	EURUSD	Exchange rate between the European euro and the U.S. dollar, it represents the lagged value EUR/USD_{t-1} and is included in the regression as an explanatory variable.
CHF/USD	CHFUSD	Exchange rate between the Swiss franc and the U.S. dollar.
GBP/USD	GBPUSD	Exchange rate between the British pound and the U.S. dollar.
RUB/USD	RUBUSD	Exchange rate between the Russian ruble and the U.S. dollar.
NZD/USD	NZDUSD	Exchange rate between the New Zealand dollar and the U.S. dollar.
BRL/USD	BRLUSD	Exchange rate between the Brazilian real and the U.S. dollar.
CNY/USD	CNYUSD	Exchange rate between the Chinese yuan and the U.S. dollar.
Gold	GOLD	Gold price, a variable that represents the commodity market and the role of gold as a safe-haven asset.
NASDAQ Composite	NASDAQ	Stock market index representing the U.S. equity market, with a strong exposure to technology companies.
Euro Stoxx 50	EUROSTOXX	Stock market index including 50 of the largest companies in the euro area.
U.S. 5-Year Treasury Yield	US5Y	Yield on U.S. government bonds with a five-year maturity.
Bitcoin	BTC/USD	Exchange rate between the cryptocurrency Bitcoin and the U.S. dollar.

Ethereum	ETH/USD	Exchange rate between the cryptocurrency Ethereum and the U.S. dollar.
Theta	THETA/USD	Exchange rate between the cryptocurrency Theta and the U.S. dollar.
USD Coin	USDC/USD	Exchange rate between the stablecoin USD Coin and the U.S. dollar.
Tether	USDT/USD	Exchange rate between the stablecoin Tether and the U.S. dollar.
Dai	DAI/USD	Exchange rate between the stablecoin Dai and the U.S. dollar.

The choice of these variables is motivated by their potential relationship with exchange rate movements and by the role played by different financial markets in the global economy. Traditional financial variables, such as exchange rates, stock indices, gold prices, and government bond yields, provide information on currency markets, financial conditions, and investor expectations. Cryptocurrencies and stablecoins are also included to account for the increasing relevance of digital assets in financial markets. In this way, the dataset combines traditional and digital financial variables and allows the forecasting models to be evaluated using an extensive set of market information.

1.3 Dataset Preparation

Before applying the forecasting models, the dataset was prepared to ensure consistency across all financial time series. This step was particularly important because the variables considered in the analysis come from different financial markets and may therefore differ in terms of trading days, missing observations, statistical properties, and scale. The preprocessing procedure was carried out before model estimation so that the different forecasting approaches could be compared using the same dataset. The first step involved collecting the historical observations for all the selected financial variables from Yahoo Finance and Investing.com. These sources provide daily market data for exchange rates, stock market indices, commodities, government bond yields, cryptocurrencies, and stablecoins. The raw data presented differences in the observation dates because the financial markets considered in the analysis do not all follow the same trading calendar.

Stock markets and government bond markets are generally closed during weekends and public holidays, whereas cryptocurrencies and stablecoins are traded continuously. As a result, some variables contained observations for days on which others did not have a corresponding value. To address this issue, the observations were aligned using a common calendar based on business days. In particular, the observations corresponding to weekends were removed from the assets that are traded continuously, so that all variables followed the same temporal structure as the traditional financial markets. This synchronization ensured that the observations included in each row of the dataset referred to the same date. As a result, each row represented one daily observation containing the available information for the target variable and for all the explanatory variables considered in the analysis. After aligning the different time series, the remaining missing observations were handled using a forward-filling procedure. With this method, when a variable did not have an observation for a particular business day, its most recent available value was carried forward until a new observation became available. This technique preserves the continuity of the time series while avoiding the unnecessary loss of observations that would result from deleting incomplete rows. At the same time, forward filling assumes that the latest available market price remains the best available estimate until new information is incorporated into the market. Before proceeding with the econometric analysis, the stationarity properties of the time series were examined. Stationarity is a crucial assumption in time-series regression, as the use of non-stationary variables can lead to spurious relationships and unreliable statistical inferences. To assess stationarity, the Augmented Dickey-Fuller (ADF) test was applied to each series in its level form. The null hypothesis of the test posits the presence of a unit root, indicating non-stationarity, while the alternative suggests stationarity. The results, reported in the following table, indicate that the majority of the series (11 out of 17) have p-values well above the conventional 5% significance level, providing strong evidence of non-stationarity. Although a few variables appeared stationary at conventional levels, the presence of non-stationarity in most series warranted a consistent transformation across all variables to ensure methodological uniformity and comparability.

Variable	ADF Statistic	p-value	Stationary
EURUSD_raw	-1,9120	0,3264	No
CNYUSD_raw	-1,5313	0,5180	No
GBPUSD_raw	-2,0611	0,2604	No
CHFUSD_raw	-0,7393	0,8363	No
RUBUSD_raw	-4,8705	0,0000	Yes
NZDUSD_raw	-2,1912	0,2094	No
BRLUSD_raw	-2,8692	0,0490	Yes
Theta Price	-4,5110	0,0002	Yes
Bitcoin Price	-1,1538	0,6932	No
Ethereum Price	-2,6511	0,0829	No
Tether Price	-6,6342	0,0000	Yes
USD Coin Price	-10,3482	0,0000	Yes
Dai Price	-6,2352	0,0000	Yes
Nasdaq Price	-0,1962	0,9390	No
Eurostoxx50 Price	-0,7261	0,8399	No
Gold Price	5,3306	1,0000	No
Treasury Yield 5Y Price	-2,0408	0,2690	No

To address the non-stationarity issue, all series were transformed into log-returns. This transformation is standard practice in financial econometrics, as it converts non-stationary price series into stationary return series, which are more suitable for regression analysis and facilitate the interpretation of coefficients in terms of percentage changes. In fact, returns are more suitable for analysing daily changes in financial variables and usually have more stable properties than price levels. For each variable, logarithmic returns were calculated as the natural logarithm of the ratio between the current price and the price observed on the previous day. The resulting values were multiplied by 100 so that the returns could be expressed in percentage terms. The next step involved checking for multicollinearity among the explanatory variables. Financial markets are highly interconnected, and several variables included in the analysis are naturally correlated. However, very high correlations can create problems in econometric models, as they can increase the variance of the estimated coefficients and make the results less stable. To investigate the problem, a pairwise correlation matrix and Variance Inflation Factors (VIF) were calculated.

Variable	VIF	Multicollinearity
EURUSD	3,8240	Low
CNYUSD	1,1268	Low
GBPUSD	3,0422	Low
CHFUSD	2,4969	Low
RUBUSD	1,0302	Low
NZDUSD	2,3984	Low
BRLUSD	1,1788	Low
Theta	2,1577	Low
Bitcoin	3,5258	Low
Ethereum	3,7418	Low
Tether	1,0453	Low
USD Coin	1,7086	Low
Dai	1,7207	Low
Nasdaq	1,4077	Low
Eurostoxx50	1,2802	Low
Gold	1,1188	Low
Treasury Yield 5Y	1,1260	Low

The VIF results were used to assess the extent of multicollinearity among the explanatory variables. However, the resulting values did not indicate severe multicollinearity among the variables retained for the regression analysis. The prediction target was then defined using the EUR/USD exchange rate. The EUR/USD series represents the dependent variable of the analysis, while the remaining financial variables were used as explanatory variables. Since the objective of the study is to forecast the following day's value, the explanatory variables were shifted by one period. In this way, the information available at date (t) was used to forecast the EUR/USD return at date (t+1). This avoids using information from the future and ensures that the model only relies on information that would have been available when the forecast was made. After defining the target and the explanatory variables, the dataset was divided chronologically into three separate subsets: training, validation, and test sets. The training set was used to estimate the models, while the validation set was used for model selection and hyperparameter tuning. In particular, the validation data helped to identify suitable model configurations and reduce the risk of overfitting. The test set was kept separate from the previous stages and was used only once the models had been finalised, in order to obtain an evaluation of their forecasting performance. Finally, standardization was applied to the explanatory variables for the

machine learning models that are affected by differences in the scale of the inputs. The StandardScaler was used for this purpose, and, to avoid information leakage, the scaler was fitted only on the training set. The mean and standard deviation obtained from the training observations were then used to transform the validation and test sets without recalculating them on those data. Therefore, no information from the validation or test periods was used during the preprocessing of the training data. This procedure ensures that the model evaluation remains consistent with the forecasting framework adopted in the study.

Variable	Mean	Median	Minimum	Maximum	Std. Dev.	Skewness	Kurtosis
EUR/USD	-0,0017	0,0000	-1,8590	2,7537	0,3993	0,2052	3,9951
CNY/USD	-0,0035	0,0000	-2,1202	2,1585	0,2895	0,4258	14,0251
GBP/USD	-0,0022	0,0000	-4,2324	3,0308	0,4471	-0,2798	8,3153
CHF/USD	0,0095	0,0000	-2,0819	4,1397	0,4263	0,9597	8,2363
RUB/USD	-0,0022	0,0000	-272,7395	271,7538	9,6827	-0,1412	689,0487
NZD/USD	-0,0108	0,0000	-4,3299	2,2092	0,5464	-0,3017	4,3007
BRL/USD	0,0039	0,0000	-3,7048	4,7429	0,7633	-0,1864	3,3581
Theta	-0,1666	-0,1338	-49,4291	25,1099	5,6923	-0,6407	6,7345
Bitcoin	0,0217	-0,0229	-17,4053	13,5764	2,9613	-0,2550	3,8942
Ethereum	0,0179	0,0347	-31,7459	22,5649	4,0065	-0,2588	5,2004
Tether	0,0000	-0,0006	-1,1360	0,9122	0,0546	-2,2463	151,5498
USD Coin	0,0000	0,0001	-2,8393	2,0951	0,0960	-8,4721	557,6326
Dai	-0,0001	0,0001	-2,5446	1,9036	0,1327	-1,5715	133,2523
NASDAQ	0,0311	0,0000	-6,1536	11,4784	1,1717	0,1066	8,3024
Eurostoxx50	0,0258	0,0000	-5,0922	7,1745	0,8963	-0,3058	7,1911
Gold	0,0595	0,0000	-10,8412	6,1647	0,9042	-1,0641	16,5363
Treasury Yield 5Y	0,0845	0,0000	-13,9502	12,8133	2,1266	0,1585	6,2372

The descriptive statistics in the table show that the mean returns are generally close to zero for most of the variables, while the median is equal or very close to zero in many cases. The standard deviation, however, varies considerably across the variables. Traditional exchange rates generally show lower volatility, whereas cryptocurrencies display substantially higher values, particularly Theta, Bitcoin and Ethereum. A notable exception is RUB/USD, which presents an extremely high standard deviation and a notably wide range between its minimum and maximum values, suggesting the presence of particularly large observations in the series. The skewness and kurtosis values also indicate that several of the distributions deviate from the normal distribution. In

particular, Tether, USD Coin and Dai show very high kurtosis values, together with pronounced negative skewness, while RUB/USD also presents a particularly high kurtosis. Overall, these results suggest the presence of fat tails and extreme observations in several of the transformed series, which is relevant when considering the statistical properties of the variables used in the following analysis.

1.4 Autocorrelation and Partial Autocorrelation Analysis

Before estimating the forecasting models, an exploratory analysis was carried out to evaluate the temporal dependence in the transformed financial series. Since the empirical analysis is based on daily logarithmic returns rather than on the original price levels, it is important to verify whether the transformation has reduced the persistence that is typically present in financial price series. For this reason, the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) were calculated for the EUR/USD return series. The ACF measures the correlation between the current value of a time series and its previous values at different lags, and it is useful for identifying whether observations are related to their past values. The PACF provides complementary information by measuring the correlation between the series and a specific lag after considering the effects of the lags between them. Looking at the ACF and PACF together therefore provides an initial indication of the serial dependence present in the series before the forecasting models are estimated. The analysis is particularly relevant in this case because the original EUR/USD price series and its returns have different statistical properties. Exchange rate levels tend to show a high degree of persistence over time and can display non-stationary behaviour, whereas daily returns are generally expected to fluctuate around a relatively stable mean. The purpose of the ACF and PACF analysis is to examine whether the transformation into returns has reduced the temporal dependence and produced a series that is more suitable for the subsequent forecasting analysis. The ACF results, shown in Figure 1, indicate that the autocorrelation coefficients are generally small and decrease rapidly as the number of lags increases. Only a limited number of lags fall outside the corresponding 95% confidence bands, suggesting that the linear dependence between the current return and its past values is relatively weak. A similar pattern can be observed in the PACF, because the significant spikes are mainly concentrated among the first few lags and become smaller at higher lags, with most of

them falling within the confidence bands. These results are also consistent with the typical behaviour observed in financial return series. In fact, daily exchange rate returns generally contain only a limited amount of linear serial dependence, as new information is incorporated into market prices relatively quickly. This implies that the amount of information contained in the past values of the EUR/USD return series is relatively limited when considering linear dependence alone. In conclusion, the ACF and PACF analysis provides evidence that the logarithmic return transformation has substantially reduced the serial dependence present in the original price series. Although some short-term dependence can still be observed at a small number of lags, the majority of the autocorrelation disappears relatively quickly.

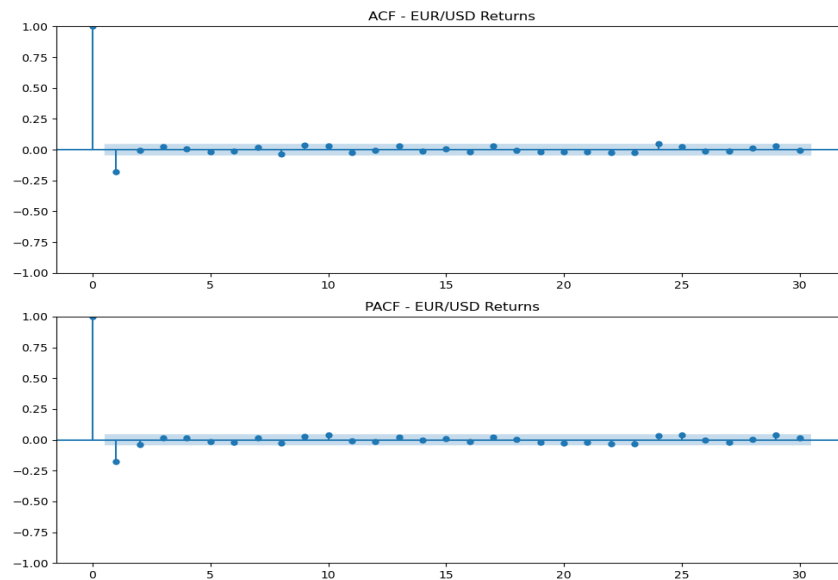


FIGURE 1 – ACF and PACF Plots

2 LINEAR REGRESSION MODELS

Linear regression models represent the first group of forecasting techniques considered in the empirical analysis. They provide a benchmark for evaluating the performance of the more flexible machine learning and deep learning models presented in the following sections. One of their main advantages is their relatively simple structure, which makes the results easier to interpret. In addition to comparing forecasting accuracy, linear models make it possible to examine the estimated relationships between the financial variables and the EUR/USD return. The linear modelling framework used in this study starts with a baseline Multiple Linear Regression estimated by Ordinary Least Squares (OLS). Two regularized regression methods, Ridge and Lasso Regression, are then considered. In addition, a backward stepwise selection procedure is applied using three different information criteria: the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and the Hannan–Quinn Information Criterion (HQC). These different specifications allow several aspects of the regression problem to be investigated, including the effects of multicollinearity, model complexity, and the selection of the most relevant explanatory variables. The dependent variable is the next-day EUR/USD return, denoted as “EURUSD_target” in the empirical implementation. The explanatory variables are based on the lagged returns of the financial variables retained after the preliminary correlation and multicollinearity analysis. More specifically, the predictors are shifted by one period, so that the information available at time (t) is used to forecast the EUR/USD return at time ($t+1$). This specification is consistent with the forecasting objective of the study and ensures that information from the future is not used when generating the predictions.

The preliminary analysis reported in the previous section showed that the return series are stationary according to the Augmented Dickey–Fuller test. The ACF and PACF analysis also indicated that the linear autocorrelation in the EUR/USD return series is generally limited. These results provide a starting point for estimating the linear regression models. However, stationarity and limited autocorrelation in the dependent variable do not by themselves guarantee that the assumptions required for statistical inference in the regression models are fully satisfied. For this reason, particular attention

is also given to the properties of the regression residuals. In financial data, heteroskedasticity and serial correlation can affect the standard errors and the statistical significance of the regression coefficients. While these issues do not necessarily make the coefficient estimates themselves invalid, they can lead to incorrect conclusions if conventional standard errors are used. To avoid that, the residuals of the estimated models are examined using the Ljung–Box test to assess the presence of serial correlation. In addition, heteroskedasticity and autocorrelation-consistent standard errors are calculated using the Newey–West procedure. These robust standard errors are used to obtain more reliable statistical inference in the presence of possible heteroskedasticity and autocorrelation.

2.1 Multiple Linear Regression

The empirical analysis starts with a Multiple Linear Regression model estimated using the Ordinary Least Squares (OLS) method. This model is used as a baseline for forecasting EUR/USD returns and provides a simple framework for analysing the relationship between the target variable and the selected financial predictors. In general, the multiple linear regression model can be written as:

$$y_{t+1} = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \varepsilon_{t+1}$$

where y_{t+1} represents the EUR/USD return to be forecasted on the following day, β_0 is the intercept, $X_{j,t}$ denotes the value of the j -th explanatory variable at time t , β_j is the corresponding regression coefficient, and ε_{t+1} is the error term. The model is estimated on the training sample using the “statsmodels” implementation. A constant term is included in the regression, and the coefficients are estimated by minimizing the sum of squared residuals. Each coefficient measures the estimated relationship between the corresponding explanatory variable and the EUR/USD return, keeping the other variables in the model constant. The OLS estimation provides the first benchmark for the forecasting analysis. The model is estimated using 1459 observations and achieves an R^2 of 0,167, while the adjusted R^2 is 0,157. This means that the explanatory variables included in the regression account for approximately 16,7% of the variation in the dependent variable within the estimation sample. Although this level of explanatory power is limited, is not unexpected in the context of daily financial return forecasting. In fact, exchange rate movements are influenced by a large number of economic and

financial factors, that cannot be fully captured by the set of variables included in the model. The overall statistical significance of the regression is nevertheless strong. The F-statistic is equal to 16,94 and its associated p-value is approximately $2,92 \times 10^{-46}$, which is below the conventional 5% significance level. Therefore, the null hypothesis that all slope coefficients are jointly equal to zero can be rejected. This result indicates that the explanatory variables contain statistically significant information about the EUR/USD return when considered together, even though the individual contribution of most variables is weak. Looking at the individual coefficients, most of the explanatory variables do not reach statistical significance at the 5% level. The coefficient associated with GBP/USD is positive and equal to 0,0618, with a p-value of 0,085. This provides some evidence of a positive relationship, even if the coefficient is not significant at the 5% level and would only be considered significant at the 10% level. From an economic perspective, the positive coefficient suggests that an increase in the lagged GBP/USD return is associated with an increase in the EUR/USD return, with the other variables constant. Since the variables are expressed as percentage returns, a one percentage point increase in the lagged GBP/USD return is associated with an estimated increase of approximately 0,062 percentage points in the EUR/USD return. The small magnitude of this coefficient is consistent with the limited short-term predictability observed in daily financial return series. The other coefficients do not provide statistically significant evidence of an individual linear relationship with the EUR/USD return at the 5% level. For example, the coefficient of the lagged EUR/USD variable is negative (-0,0503), but its p-value of 0,267 indicates that the estimated relationship is not statistically distinguishable from zero. Similarly, CNY/USD has a negative coefficient of -0,0164 and a p-value of 0,612, while CHF/USD has a positive coefficient of 0,0055 with a p-value of 0,879. The coefficients associated with RUB/USD, NZD/USD and BRL/USD are also statistically insignificant. The cryptocurrency variables show similarly limited effects. For example, Bitcoin has a coefficient extremely close to zero ($7,633 \times 10^{-5}$) and a p-value of 0,990. Ethereum has a coefficient of 0,0033 and a p-value of 0,465, while Theta has a coefficient of 0,0028 and a p-value of 0,232. So, none of these cryptocurrency variables shows significant evidence of an individual linear relationship with the EUR/USD return. The difference between the strong joint significance of the model and the limited individual significance of most regressors is worth nothing. The highly

significant F-test indicates that the explanatory variables contain information that is relevant when considered jointly. At the same time, the individual t-tests suggest that it is difficult to identify a strong and precise effect for most individual predictors. This result partly reflects the high degree of interdependence between financial markets, as well as the weak individual predictability of daily exchange rate returns. The results of the OLS regression are shown in the following table.

Variable	Coefficient	Std Error	t-statistic	p-value
Eurostoxx50	0,0558	0,0122	4,5681	0,0000
Nasdaq	0,0414	0,0100	4,1606	0,0000
Treasury Yield 5Y	-0,0207	0,0046	-4,4933	0,0000
Gold	0,1185	0,0143	8,2945	0,0000
Dai	0,1629	0,0850	1,9168	0,0555
GBPUSD	0,0618	0,0359	1,7216	0,0854
RUBUSD	0,0014	0,0009	1,5316	0,1258
BRLUSD	0,0194	0,0130	1,4887	0,1368
Theta	0,0028	0,0024	1,1961	0,2319
EURUSD	-0,0503	0,0453	-1,1109	0,2668
Ethereum	0,0033	0,0045	0,7303	0,4653
USD Coin	-0,0601	0,1169	-0,5146	0,6069
CNYUSD	-0,0164	0,0323	-0,5069	0,6123
Tether	-0,0672	0,1643	-0,4090	0,6826
NZDUSD	-0,0055	0,0272	-0,2017	0,8402
CHFUSD	0,0055	0,0361	0,1528	0,8786
Bitcoin	0,0001	0,0059	0,0130	0,9896

The out-of-sample results provide a more conservative assessment of the model's forecasting ability. On the test set, the OLS model obtains an RMSE of 0,4036 and an MAE of 0,2607, while the test-set R^2 is only 0,0218. However, the model explains approximately 2,18% of the variation in the EUR/USD returns that were not used for estimation. The difference between the in-sample R^2 of 16,7% and the out-of-sample R^2 of 2,2% suggests that the relationships, identified during estimation, have limited predictive power when the model is applied to unseen observations. As it is shown in Figure 2, the predicted series generally remains relatively close to zero and follows the central tendency of the EUR/USD returns. In contrast, the actual series shows considerably larger positive and negative fluctuations. However, several of the more pronounced movements in the actual returns are not adequately captured by the OLS

forecasts. This behaviour explains why the model can achieve a low average prediction error while still producing a low R^2 : the regression is able to capture part of the general behaviour of returns, but it has limited ability to anticipate the larger movements.

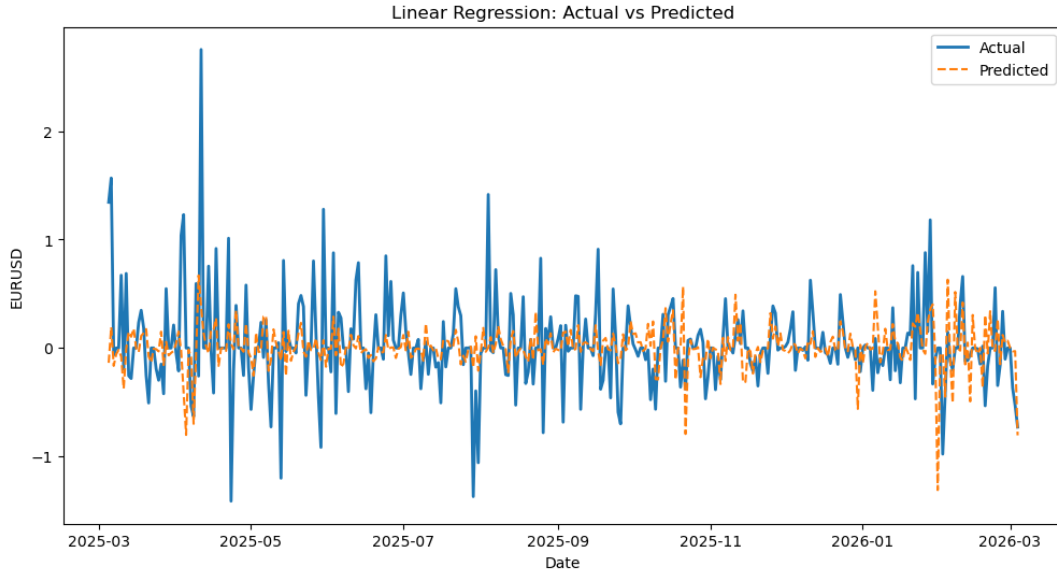


FIGURE 2- Linear Regression (Actual vs Predicted)

In conclusion, the OLS model shows that the linear specification contains some relevant information about EUR/USD returns, especially when the explanatory variables are considered jointly. However, the lower R^2 obtained on the test set indicates that this relationship is not strong enough to provide accurate forecasts for unseen observations. In fact, the results suggest that a linear model has limited ability to capture the short-term movements of the EUR/USD exchange rate. This represents an important motivation for considering more flexible models in the following sections and chapters.

2.1.1 Residual Diagnostics

After estimating the OLS model, the residuals were examined to check whether they still contain serial correlation. This diagnostic step is important because the presence of autocorrelation in the residuals would indicate that part of the temporal structure of the dependent variable remains unexplained by the model. In that case, the assumption of uncorrelated errors required for standard OLS inference would not be satisfied.

The Ljung–Box test was used for this purpose. The test examines whether the residual autocorrelations are jointly equal to zero up to a given number of lags. More formally, the null hypothesis is expressed as

$$H_0: \rho_1 = \rho_2 = \dots = \rho_h = 0$$

where ρ_j represents the autocorrelation of the residuals at lag j . The alternative hypothesis states that at least one of these autocorrelations is different from zero. Moreover, a p-value below the chosen significance level provides evidence against the null hypothesis, while a higher p-value means that the null hypothesis cannot be rejected and that no significant residual autocorrelation is detected up to the selected lag. The Ljung–Box test complements the ACF and PACF analysis presented previously. While the ACF and PACF were used to examine the serial dependence of the EUR/USD return series before estimating the models, the Ljung–Box test is applied to the residuals obtained from the OLS regression. This distinction is important because the purpose of the residual analysis is to determine whether the model has successfully captured the linear temporal dependence present in the data. The test was performed at lags 5, 10, and 20, and the results are reported in the table below. At lag 5, the Ljung–Box statistic is 1,7631, with a p-value of 0,8809. Since the p-value is above the 5% significance level, the null hypothesis cannot be rejected. The same result is obtained at lag 10, where the test statistic is 8,7029 and the p-value is 0,5605. At lag 20, the statistic increases to 19,8153, but the corresponding p-value remains relatively high at 0,4695. Once again, there is no sufficient evidence to reject the null hypothesis. In conclusion, the Ljung–Box results do not indicate statistically significant autocorrelation in the OLS residuals at any of the three lag lengths considered. This suggests that the prediction errors do not follow a clear linear pattern over time. In particular, the residuals do not appear to contain additional serial information that could be captured by simply adding autoregressive terms to the linear specification. The test therefore provides evidence that the OLS model does not leave substantial linear serial correlation unexplained.

lb_stat	lb_pvalue	Conclusion
1,7631	0,8809	No autocorrelation
8,7029	0,5605	No autocorrelation
19,8153	0,4695	No autocorrelation

2.1.2 Stepwise Regression for OLS

To obtain a more parsimonious specification, a backward stepwise regression procedure was applied to the full OLS model. The procedure was performed separately using three different information criteria: the Akaike Information Criterion (AIC), the Bayesian

Information Criterion (BIC), and the Hannan–Quinn Information Criterion (HQC). In each case, the procedure started from the full set of explanatory variables and progressively removed variables when their exclusion improved the value of the selected criterion. Comparing the resulting models makes it possible to assess whether the choice of the information criterion affects the variables retained in the final specification. The three procedures lead to models with different levels of complexity. Using the AIC as the selection criterion, eight explanatory variables are retained: RUB/USD, BRL/USD, THETA, DAI, NASDAQ, Euro Stoxx 50, Gold, and the U.S. 5-Year Treasury Yield. The resulting specification has an AIC of 1200,041, while its BIC and HQC values are 1247,611 and 1217,787, respectively. Compared with the other two procedures, the AIC retains the largest number of variables, resulting in the least parsimonious specification.

The BIC procedure leads to a more restricted model, retaining only four explanatory variables: NASDAQ, Euro Stoxx 50, Gold, and the U.S. 5-Year Treasury Yield. For this specification, the AIC is 1207,992, the BIC is 1234,419, and the HQC is 1217,851. The smaller number of regressors is consistent with the stronger penalty that the BIC applies to model complexity. As a result, the BIC places greater emphasis on obtaining a simpler specification and retains only four variables from the original set. The HQC procedure produces an intermediate specification with six explanatory variables: THETA, DAI, NASDAQ, Euro Stoxx 50, Gold, and the U.S. 5-Year Treasury Yield. The corresponding AIC, BIC, and HQC values are 1201,221, 1238,219, and 1215,023, respectively. In terms of the number of variables, the HQC specification therefore falls between the AIC and BIC models. This is also consistent with the different penalties for model complexity associated with the three criteria. The comparison of the selected variables provides some interesting results, because NASDAQ, Euro Stoxx 50, Gold, and the U.S. 5-Year Treasury Yield are retained by all three procedures. These variables can be considered the most consistent predictors within the linear modelling framework, as their selection does not depend on the information criterion used. On the other hand, RUB/USD and BRL/USD are retained only when using the AIC. THETA and DAI are selected by both the AIC and HQC procedures but are excluded when the BIC is used. This difference suggests that these variables provide enough additional information to justify their inclusion under the less restrictive AIC and HQC criteria, but not under the stronger penalty imposed by the BIC. The information criteria also illustrate the different

objectives of the three selection procedures. The AIC achieves the lowest AIC value, equal to 1200,041, while the BIC specification has the lowest BIC value, 1234,419. Similarly, the HQC has the lowest HQC value at 1215,023. This confirms that each stepwise procedure selects the model that provides the best trade-off between model fit and complexity according to its own criterion. From an economic perspective, the variables that are consistently selected by all three procedures represent different aspects of financial market conditions. NASDAQ and the Euro Stoxx 50 capture movements in major equity markets, while Gold represents an important commodity and is also commonly associated with safe-haven demand. The U.S. 5-Year Treasury Yield provides information about the interest-rate environment and financial conditions in the United States. Their consistent selection across the three procedures suggests that these variables contain information that is relevant to the EUR/USD return within the linear modelling framework. In conclusion, the stepwise analysis shows that the full OLS model can be reduced while retaining a smaller group of variables that are selected by the different information criteria. The AIC produces the least restrictive specification with eight regressors, while the BIC selects a more parsimonious model with only four variables. The HQC lies between the two, retaining six explanatory variables. The fact that NASDAQ, Euro Stoxx 50, Gold, and the U.S. 5-Year Treasury Yield appear in all three specifications provides some evidence that these variables are robust to the choice of model selection criterion.

2.1.3 HAC / Newey–West Robust Standard Errors

Heteroskedasticity and Autocorrelation Consistent (HAC) standard errors were also used to account for possible heteroskedasticity and serial correlation in the regression errors. The HAC estimator, commonly referred to as the Newey–West procedure, does not change the OLS coefficient estimates or the fitted values of the model. It only affects the estimated covariance matrix and, consequently, the standard errors, t-statistics and p-values used to assess statistical significance. In this analysis, HAC standard errors were calculated using five lags. The comparison between conventional OLS and HAC results shows that the estimated coefficients remain unchanged, while the significance of some variables changes. The results confirm that most of the main financial variables remain significant after applying the HAC correction. In particular, NASDAQ, Euro Stoxx 50, Gold and the U.S. 5-Year Treasury Yield remain statistically significant, with HAC p-

values of approximately 0,0007, 0,0029, below 0,001, and 0,0021, respectively. Therefore, the statistical relevance of these variables is not driven by the use of conventional standard errors and remains robust to the HAC correction. One of the more noticeable changes regards the RUB/USD exchange rate. Under conventional OLS standard errors, its coefficient is not statistically significant at the 5% level, with a p-value of approximately 0,126. After applying the HAC correction, its p-value decreases to 0,002, making the coefficient significant at the 1% level. The estimated coefficient is positive and equal to approximately 0,0014, indicating a positive association between RUB/USD and the EUR/USD target, conditional on the other regressors included in the model. A similar change can be observed for DAI, whose p-value decreases from around 0,055 under conventional OLS inference to 0,028 under HAC standard errors. Consequently, DAI becomes statistically significant at the 5% level after the robust correction. Its estimated coefficient is more or less 0,163, indicating a positive conditional association with the EUR/USD target. For the remaining variables, the HAC correction does not substantially change the conclusions regarding significance. EUR/USD, CNY/USD, GBP/USD, CHF/USD, NZD/USD, BRL/USD, THETA, Bitcoin, Ethereum, Tether and USD Coin remain statistically insignificant at the 5% level. In particular, GBP/USD remains relatively close to the significance threshold, with a HAC p-value of approximately 0,077, but it remains above the 5% level. The overall explanatory power of the regression is unchanged by the correction. The model continues to report an R^2 of 0,167 and an adjusted R^2 of 0,157. This is expected because HAC estimation affects the covariance matrix and the resulting statistical inference, rather than the OLS coefficient estimates or the fitted values. Moreover, the results indicate that approximately 16,7% of the variation in the EUR/USD target is explained by the regressors included in the model, while the remaining variation reflects factors not captured by the linear specification. In conclusion, the HAC results provide a more robust basis for evaluating the statistical significance of the estimated coefficients. The fact that NASDAQ, Euro Stoxx 50, Gold and the U.S. 5-Year Treasury Yield remain significant suggests that their relationship with the EUR/USD target is not dependent on the use of conventional standard errors. At the same time, the changes observed for RUB/USD and DAI show that the choice of standard errors can affect the statistical conclusions, particularly when working with financial time-series data.

2.2 Ridge Regression

To complement the OLS specification, two regularized linear regression methods were considered: Ridge Regression and Lasso Regression. Both methods extend the standard linear regression framework by introducing a penalty term to the objective function. The main purpose of regularization is to reduce the magnitude of the estimated coefficients and, as a result, limit the risk of overfitting. This can be particularly useful when the number of explanatory variables is large or when some regressors contain overlapping information. Before estimating the regularized models, the explanatory variables were standardized using “StandardScaler”. This step is important for both Ridge and Lasso because the regularization penalty depends directly on the magnitude of the coefficients. If the variables were left on different scales, they could be penalized differently because of their units of measurement rather than because of their actual contribution to the model. In the empirical analysis, the scaler was fitted only on the training observations and was then applied unchanged to the validation and test sets. This was done to prevent information from future observations from entering the model estimation process. Ridge Regression introduces an L2 penalty into the OLS objective function. The estimator can be expressed as

$$\hat{\beta}^{Ridge} = \arg \min_{\beta} \left[\sum_{t=1}^T (y_t - x'_t \beta)^2 + \alpha \sum_{j=1}^p \beta_j^2 \right]$$

where y_t represents the EUR/USD target, x_t denotes the vector of explanatory variables, and α is the regularization parameter controlling the strength of the penalty. When $\alpha = 0$, the Ridge model becomes equivalent to the standard OLS estimator. As α increases, larger coefficients are increasingly penalized and are shrunk towards zero. Unlike Lasso, however, Ridge generally does not force coefficients to become exactly zero. This means that all explanatory variables can remain in the model, while their influence is reduced depending on the amount of regularization applied. So, instead of completely excluding correlated variables, the method distributes their contribution across the regression coefficients. This can lead to more stable estimates compared with those obtained from unrestricted OLS, especially when some of the explanatory variables contain similar or overlapping information. In our case, the optimal value obtained was $\alpha = 95,4548$. This high value indicates that a substantial degree of regularization was selected, suggesting

that shrinking the coefficients was beneficial for the out-of-sample predictive performance of the model. Ridge Regression achieved a MSE of 0,1614, corresponding to an RMSE of 0,4018, while the MAE was 0,2586, and the resulting R^2 was 0,0308. These results indicate that the model explains only a limited part of the variation in the EUR/USD target, which is also shown in the relatively low R^2 . However, the predictive performance is slightly better than the one obtained with the OLS model. In fact, the OLS specification produced an RMSE of 0,4036, an MAE of approximately 0,2607 and an R^2 of around 0,0218; Ridge therefore produces a modest improvement in all three test-set metrics. Even though the differences are relatively small, the results suggest that the regularization mechanism contributes to more stable out-of-sample predictions. The effect of regularization can also be observed in the Ridge coefficient paths in Figure 3. As α increases, the estimated coefficients progressively move towards zero. This is the effect of the L2 penalty: rather than removing individual variables from the model, Ridge reduces the magnitude of their coefficients. At the selected value of α , the coefficients are considerably more constrained than under an unregularized linear specification. This behaviour is particularly relevant in the present dataset, where several financial variables may contain overlapping information. By shrinking the coefficients, Ridge reduces the influence of potentially unstable individual estimates while retaining all predictors in the model. In this sense, the regularization does not change the set of explanatory variables considered, but instead controls the extent to which each of them contributes to the final prediction.

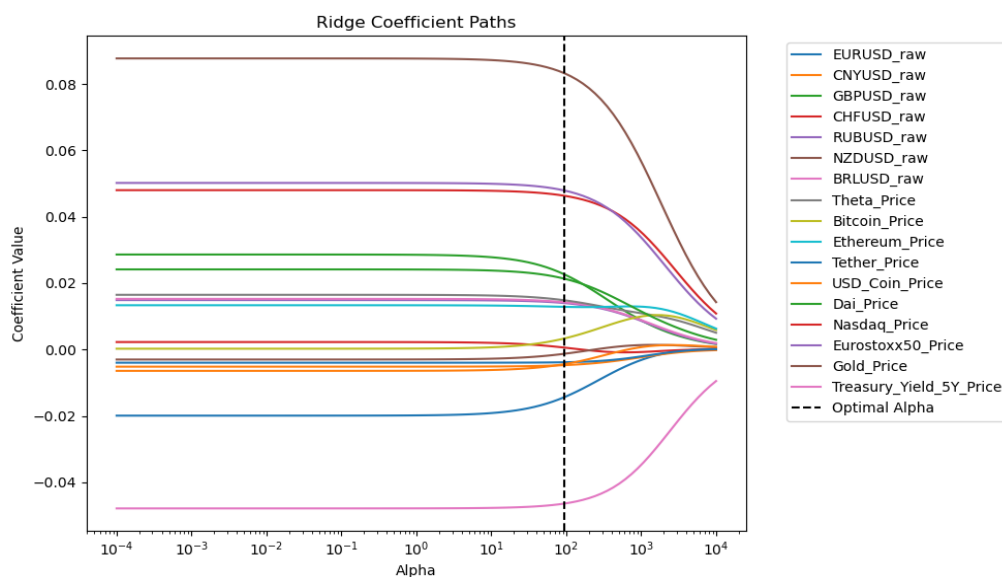


FIGURE 3 – Ridge Coefficient Paths

The predictive performance of the Ridge regression model is illustrated in Figure 4, which displays the actual EUR/USD returns alongside the model's predictions over the test period. The plot reveals that the Ridge model generally follows the direction and magnitude of the actual returns, capturing the overall trend of the series. However, consistent with the nature of L2 regularization, the Ridge predictions appear somewhat smoothed compared to the actual data. This is particularly evident during periods of heightened volatility, where the model tends to understate the amplitude of extreme movements. This smoothing effect is a direct consequence of the penalty applied to the coefficients, which reduces their variance at the cost of introducing a slight bias. As a result, the Ridge model is less sensitive to short-term fluctuations, which can be advantageous for reducing overfitting, but may also lead to a slight underestimation of extreme market movements.

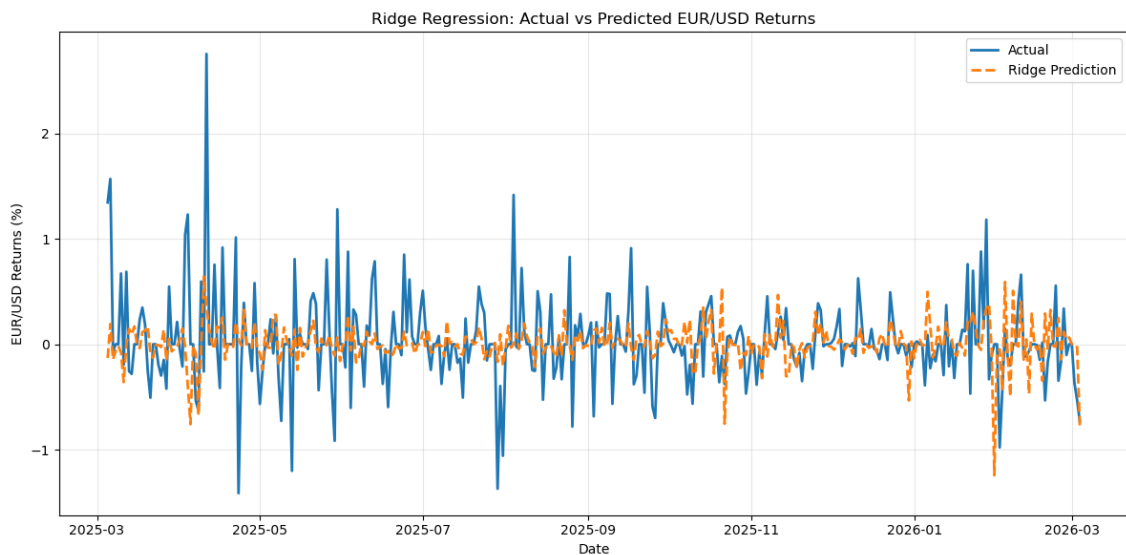


FIGURE 4 – Ridge Regression (Actual Vs Predicted)

The residuals were also examined using the Ljung–Box test at lags 5, 10 and 20. The resulting $-values$ were 0,9045, 0,5704 and 0,4912, respectively. Since all values are above the 5% significance level, the null hypothesis of no residual autocorrelation cannot be rejected at any of the three lag lengths. Therefore, there is no statistical evidence of residual autocorrelation in the Ridge model. To conclude, the Ridge specification provides a small but measurable improvement over OLS in terms of test-set predictive performance, while maintaining the complete set of explanatory variables. Its main advantage in this application therefore appears to be related to the stabilization of the coefficient estimates rather than to a substantial increase in the proportion of variance

explained. The relatively high value of the selected regularization parameter and the improvement observed across the three test-set metrics suggest that some degree of coefficient shrinkage is useful for this dataset, even though the overall explanatory power of the model remains limited.

2.3 Lasso regression

As a further regularization model, the Least Absolute Shrinkage and Selection Operator (Lasso) regression was estimated to assess whether a more parsimonious specification could be obtained by simultaneously controlling for multicollinearity and performing variable selection. Unlike Ridge regression, which shrinks the estimated coefficients towards zero without eliminating variables, Lasso introduces an L_1 penalty that can force some coefficients to become equal to zero. The estimator can be expressed as:

$$\hat{\beta}^{Lasso} = \arg \min_{\beta} \left[\sum_{t=1}^T (y_t - x'_t \beta)^2 + \alpha \sum_{j=1}^p |\beta_j| \right]$$

where y_t represents the EUR/USD target, x_t denotes the vector of explanatory variables, and α controls the strength of the regularization. As in the Ridge specification, the first term corresponds to the usual OLS sum of squared residuals, while the second term represents the regularization penalty. The difference is that Lasso penalizes the absolute value of the coefficients, rather than their squared values as in Ridge. This particular form of the penalty allows some coefficients to be reduced to zero when the regularization is sufficiently strong. This property is particularly useful in the present setting, given the relatively large number of financial and cryptocurrency variables considered as potential predictors of the EUR/USD exchange rate. By setting some coefficients to zero, Lasso can reduce the number of variables included in the final specification while retaining those that provide greater linear predictive information. The regularization parameter was selected through cross-validation. The resulting optimal value of α was 0,0073, corresponding to the minimum of the cross-validated mean squared error. As shown by the cross-validation curve in Figure 5, the error initially decreases as the penalty increases, reaches a minimum around the selected value of α , and then increases again when stronger regularization is applied. So, the selected value represents a balance between model flexibility and coefficient shrinkage. In particular, it avoids both a penalty that is

too weak to provide a meaningful regularization effect and excessive regularization that could remove useful information from the model.

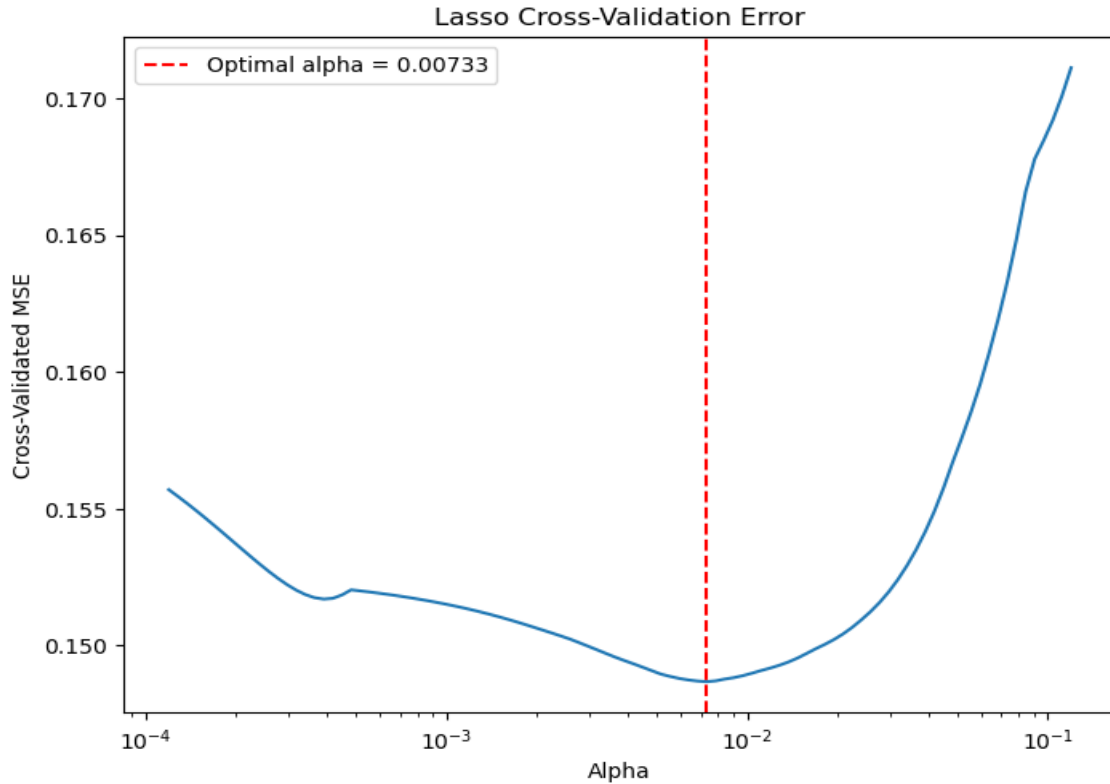


FIGURE 5 – Lasso Cross-Validation Curve

The Lasso model was then evaluated on the test set. The resulting MSE is 0,1612, while the RMSE is 0,4015 and the MAE is 0,2574. The coefficient of determination is $R^2 = 0,0323$. Therefore, the model explains approximately 3,2% of the out-of-sample variation in the EUR/USD target. Although the explanatory power remains very limited, the results are consistent with the characteristics of exchange-rate prediction, where short-term movements can be difficult to explain using a linear specification based only on the available predictors. In particular, the relatively low R^2 means that the Lasso model should not be interpreted as providing a deterministic explanation of EUR/USD movements. Rather, its main contribution in this analysis is to identify the variables that contain the greatest amount of linear predictive information in the set of regressors considered. Figure 6 presents the actual EUR/USD returns alongside the predictions generated by the Lasso regression model over the same test period. Similar to the Ridge model, the Lasso predictions broadly track the general direction of the actual returns,

demonstrating the model's ability to capture the underlying trend of the series. A notable feature of the Lasso predictions is their slightly sparser structure. This is a direct consequence of the L1 regularization, which tends to shrink the coefficients of less influential variables exactly to zero, effectively performing variable selection. As a result, the Lasso model may produce predictions that are marginally more responsive to the most relevant predictors, while disregarding noise from irrelevant variables.

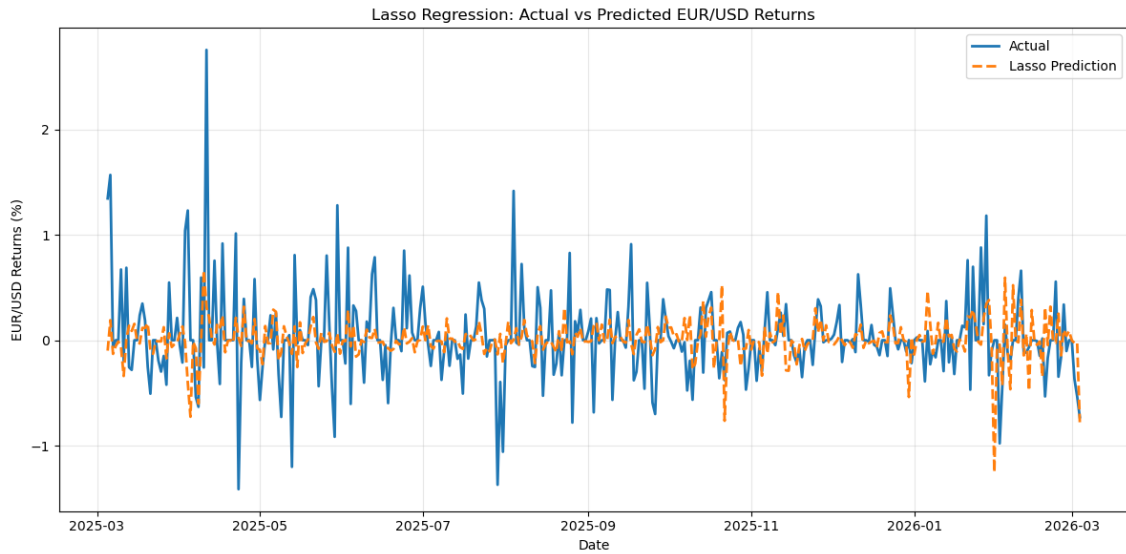


FIGURE 6 – Lasso Regression (Actual Vs Predicted)

Moreover, as shown in Figure 7, out of the 17 variables initially considered, the model assigns non-zero coefficients to only 10 variables, while the remaining 7 coefficients are set equal to zero. The parameters retained by the model are Gold Price, Nasdaq Price, Eurostoxx50 Price, DAI Price, Ethereum Price, Theta Price, RUB/USD, BRL/USD, GBP/USD, and the 5-year Treasury yield. On the other hand, the coefficients associated with EUR/USD, CNY/USD, CHF/USD, NZD/USD, Bitcoin, Tether, and USD Coin are reduced to zero. This provides an example of the variable selection property of Lasso and suggests that, within the selected model, these variables do not provide sufficient additional linear predictive information to justify their inclusion. The magnitude and sign of the retained coefficients also provide some information about the relationships identified by the model. Among the positive coefficients, Gold Price has the largest estimated coefficient, equal to approximately 0,084, followed by Nasdaq Price (0,046), Eurostoxx50 Price (0,045), and DAI Price (0,013). The largest negative

coefficient is associated with the 5-year Treasury yield, with a value of $-0,041$. The remaining selected variables have comparatively smaller coefficients. The coefficient structure also highlights a difference with respect to the unrestricted OLS specification. While the OLS model retains all regressors and estimates an individual coefficient for each of them, Lasso applies a penalty that reduces some coefficients exactly to zero and removes a subset of variables from the final specification. In this sense, the Lasso results complement the OLS and Ridge analyses. OLS provides the benchmark unrestricted linear specification, Ridge addresses potential coefficient instability through continuous shrinkage, while Lasso goes one step further by performing a selection of the predictors that remain in the model. Finally, the residuals of the model were examined using the Ljung–Box test at lags 5, 10, and 20. The corresponding p-values are 0,887, 0,562, and 0,482, respectively. Since all values are substantially above the 5% significance level, the null hypothesis of no correlation cannot be rejected at any of the considered lags. Therefore, there is no statistical evidence of residual autocorrelation in the Lasso model. The absence of significant autocorrelation suggests that this specification is not leaving an evident time-dependent structure unexplained, despite its modest out-of-sample explanatory power. In conclusion, the Lasso results show that regularization can simplify the linear forecasting specification while retaining a limited set of informative predictors. At the same time, the low R^2 confirms that the relationship between the EUR/USD exchange rate and the financial variables considered is only partially captured by a linear model. The main contribution of Lasso in this analysis therefore lies in providing a more parsimonious specification and identifying a smaller group of variables that appear to contain linear predictive information.

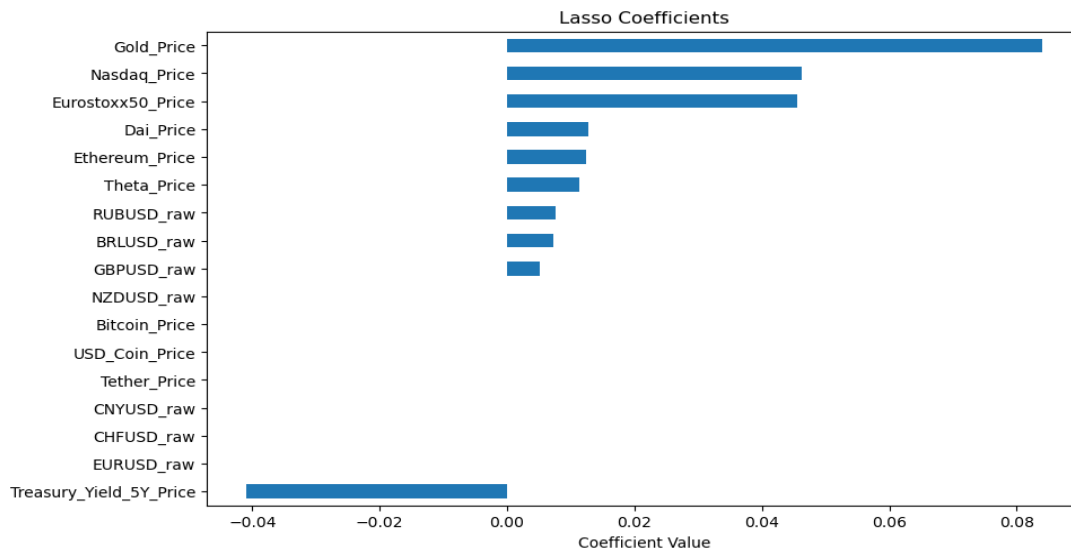


FIGURE 7 – Lasso Coefficients

2.4 Comparison of the Linear Models

To conclude the analysis of the linear models, the out-of-sample performance of the three estimated specifications was compared using the same evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). The comparison includes the baseline Ordinary Least Squares (OLS) model and the two regularized approaches, Ridge and Lasso regression. Since all three models are evaluated on the same test set, the resulting metrics provide a direct comparison of their predictive performance. The results show a gradual improvement when regularization is introduced.

The OLS model obtains an MSE of 0,1629, an RMSE of 0,4036, and an MAE of 0,2607, with an R^2 equal to 0,0218. These values represent the benchmark for evaluating the two regularized models. The low R^2 confirms that the linear relationship between the EUR/USD target and the set of explanatory variables captures only a limited proportion of the variation in the dependent variable. Despite the limited predictive power, OLS remains an important model, since it estimates the relationships between the variables without imposing any regularization penalty.

The Ridge regression produces a modest improvement over OLS across all four metrics. Its MSE decreases to 0,1614, the RMSE falls to 0,4018 and the MAE to 0,2586. At the same time, the R^2 increases from 0,0218 to 0,0308. Even if these improvements are small

in absolute terms, they suggest that the introduction of an L_2 penalty provides some benefit in terms of out-of-sample generalization. By shrinking the coefficients towards zero, Ridge reduces the influence of potentially unstable estimates and can provide a more robust specification when some predictors contain correlated or overlapping information. The Lasso model achieves the best overall performance among the three linear specifications. Its MSE is 0,1612, the lowest of the three models, while the RMSE is 0,4015 and the MAE is 0,2574. The R^2 reaches 0,0323, representing the highest explanatory power obtained among the linear models considered. The differences in predictive accuracy remain very small, but the improvement across all four evaluation metrics indicates that Lasso provides the strongest out-of-sample performance in this comparison. Overall, the results suggest that the two regularized models perform slightly better than the unrestricted OLS specification on the test set, with Lasso obtaining the best results according to all the metrics considered. The progression from OLS to Ridge and then to Lasso reflects the different degree and type of regularization introduced by the three approaches. OLS provides the unrestricted benchmark, Ridge improves coefficient stability through continuous shrinkage, while Lasso combines shrinkage with explicit variable selection. At the same time, the low R^2 values across all three models should be taken into account when interpreting these results. Even the best-performing specification explains only approximately 3,2% of the out-of-sample variation in the EUR/USD target. Therefore, although regularization leads to a measurable improvement compared with the baseline model, the results indicate that the dynamics of the exchange rate cannot be adequately captured by a simple linear relationship alone. This provides a natural motivation for the following analysis of more flexible modelling approaches, where nonlinear relationships and more complex dependencies can be considered.

3 NON-LINEAR PARAMETRIC MODELS

The linear models presented in the previous chapter provide a useful benchmark for analysing the relationship between the target variable and the set of explanatory variables considered in this study. However, these models assume that the effect of each predictor on the dependent variable can be adequately described through a linear relationship. Although regularization methods such as Ridge and Lasso regression can improve model stability and, in the case of Lasso, perform variable selection, they do not change this basic assumption about the functional form of the model. In financial markets, relationships between variables can be more complex than those captured by a simple linear specification. The effect of a predictor may change depending on its level, while interactions between variables or other nonlinear patterns may arise under different market conditions. For this reason, the analysis is extended in this chapter to models that provide greater flexibility in describing the relationship between the explanatory variables and the EUR/USD exchange rate. The first approach considered is Polynomial Regression, which extends the standard linear regression framework by introducing polynomial transformations of the explanatory variables. This allows the model to capture nonlinear relationships while maintaining a simple and interpretable structure. The Polynomial Regression model is evaluated using the same general out-of-sample framework adopted throughout the empirical analysis. Its predictive performance is assessed using standard error metrics, while the residuals are also examined to determine whether the model adequately captures the temporal structure of the target variable. This provides a basis for comparing the results of the nonlinear specification with those obtained from the linear models.

3.1 Polynomial Regression

To investigate whether nonlinear relationships between the explanatory variables and the target variable could improve the predictive performance of the models considered so far, a Polynomial Regression model was also implemented. Unlike standard linear regression, Polynomial Regression extends the feature space by introducing polynomial transformations of the original predictors and their interactions. In this way, the model can capture nonlinear patterns that cannot be represented by a linear specification. The

performance of the model was evaluated using the same metrics adopted for the previous models: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). The resulting values were 0,2644 for MSE, 0,5142 for RMSE, 0,3009 for MAE, and $-0,5875$ for R^2 . These results show that the Polynomial Regression does not improve the predictive performance obtained with the linear specifications. In particular, both the MSE and RMSE are substantially higher than those obtained with Linear Regression, Ridge and Lasso. The MAE is also higher, indicating that the polynomial specification produces larger prediction errors on average. The most relevant result is the negative R^2 value of approximately $-0,588$. A negative R^2 on the test set indicates that the polynomial model performs worse than a simple benchmark based on predicting the mean of the dependent variable. Therefore, the additional nonlinear structure introduced by the polynomial transformation does not appear to provide useful information for predicting the EUR/USD target in the considered dataset. The graphical comparison between actual and predicted values grants further evidence of the model's limited predictive ability. As shown in Figure 8, the polynomial predictions tend to remain concentrated around the central part of the target distribution and are able to reproduce some of the general fluctuations in the series. However, they do not adequately capture several of the larger positive and negative movements observed in the actual EUR/USD series. This becomes evident during periods characterized by extreme observations, when the difference between the actual and predicted values becomes larger. The polynomial model therefore appears to have difficulty reproducing the high variability and occasional extreme movements that characterize the target series. The poor performance of the polynomial specification may also be related to the characteristics of financial time series. Although nonlinear relationships may exist between exchange rates, financial assets, and macroeconomic variables, simply introducing polynomial terms does not necessarily provide an effective way of capturing these relationships. In a dataset characterized by substantial volatility and weak linear predictability, increasing the flexibility of the functional form can instead result in a less stable model and poorer out-of-sample performance. Another possible explanation is the increase in the complexity of the feature space produced by the polynomial expansion. By adding polynomial and interaction terms, the model has more flexibility to fit relationships that are present in the training sample but may not be maintained in unseen

observations. This can reduce the model's ability to generalize to the test set and may therefore contribute to the poorer predictive results observed in this case. Overall, the results indicate that the introduction of polynomial terms does not improve the predictive performance for the EUR/USD series in this application. Compared with the previously estimated Linear Regression, Ridge and Lasso models, Polynomial Regression produces larger prediction errors and a substantially lower R^2 . The results therefore suggest that simply increasing the flexibility of a linear model through polynomial terms is not sufficient to capture the dynamics of the target variable. For this reason, the subsequent analysis considers other nonlinear approaches that can model more complex relationships without relying only on polynomial transformations.

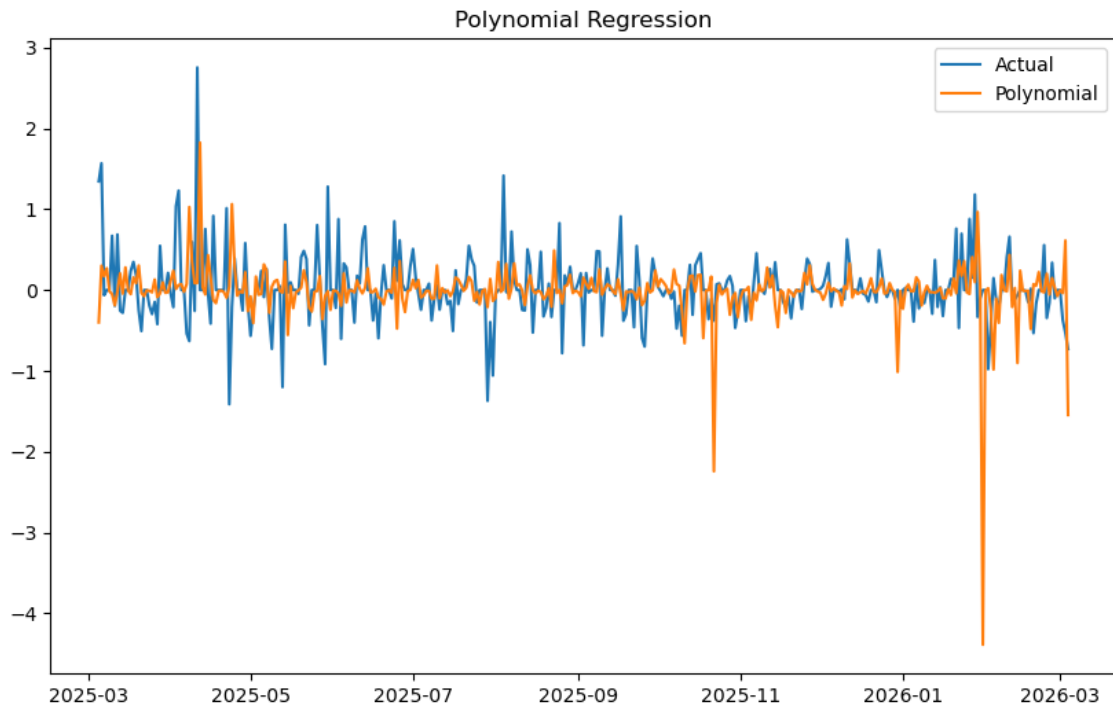


FIGURE 8 – Polynomial Regression (Actual vs Predicted)

3.2 Spline Regression

Spline regression represents an alternative approach to modelling nonlinear relationships. Rather than approximating the entire relationship between the dependent and explanatory variables with a single polynomial function, spline methods divide the range of the predictors into different regions using a set of knots. A polynomial function is then fitted within each region, while continuity and smoothness conditions are imposed at the knots. This structure allows the model to capture more complex nonlinear patterns while

reducing some of the instability that can occur when high-degree global polynomials are used. The spline model was estimated using the same empirical framework adopted for the other predictive models. In particular, the chronological structure of the dataset was preserved, and the model was evaluated on the same independent test set used for the linear, Ridge, Lasso and Polynomial specifications. Using the same test set makes it possible to compare the predictive performance of the different models under comparable out-of-sample conditions and avoids attributing differences in performance to changes in the evaluation sample. The spline model achieves an MSE of 0,1515, corresponding to an RMSE of 0,3892, while the MAE is equal to 0,2532. The resulting R^2 is 0,0905. These values represent a clear improvement compared with the Polynomial specification considered previously. The spline model produces substantially lower prediction errors and, unlike the Polynomial model, achieves a positive out-of-sample R^2 . The comparison between the actual and predicted EUR/USD values also provides some information about how the model behaves. As shown in Figure 9, the spline predictions generally remain concentrated around the central region of the target series and follow the broad movements of the actual observations. The model is able to reproduce a considerable part of the regular variation in the series, particularly when the EUR/USD values remain relatively close to zero. However, the graph also shows that the spline predictions are considerably smoother than the actual series and tend to underestimate the magnitude of some of the largest positive and negative movements. This behaviour is particularly visible during periods characterized by pronounced spikes in the exchange rate series. Such deviations are not unexpected in a financial forecasting context, since extreme movements can be difficult to predict using contemporaneous explanatory variables alone. From a predictive perspective, the spline specification represents the best-performing model among those considered so far. Its MSE of 0,1515 is lower than the values obtained by the linear regression model, Ridge regression, Lasso regression, and Polynomial regression. The same ranking can be observed for the RMSE and MAE measures: the spline model records an RMSE of 0,3892 and an MAE of 0,2532, both lower than the corresponding values of the other specifications. The improvement is particularly clear when compared with Polynomial regression. This suggests that introducing nonlinearity through a more locally flexible functional form is considerably more effective in this case than using a single global polynomial relationship. The R^2

value provides further evidence in favour of the spline specification. Although the model does not explain a large proportion of the total variability in the EUR/USD target, its R^2 of approximately 9,0% is higher than the values obtained by the linear models and is also considerably better than the negative value observed for Polynomial regression. This means that the additional nonlinear flexibility introduced by the spline model allows it to capture part of the systematic variation in the target that is not adequately represented by the other specifications. In conclusion, the spline regression provides an important result for the empirical analysis. Compared with the linear specifications, it allows for greater flexibility without experiencing the severe deterioration in predictive performance observed with Polynomial regression. The combination of the lowest MSE, RMSE and MAE among the models examined so far, together with the highest R^2 , indicates that the spline specification offers the most effective representation of the nonlinear relationships considered at this stage of the analysis. This result motivates the continuation of the empirical analysis towards more flexible and potentially more powerful nonlinear and machine-learning approaches.

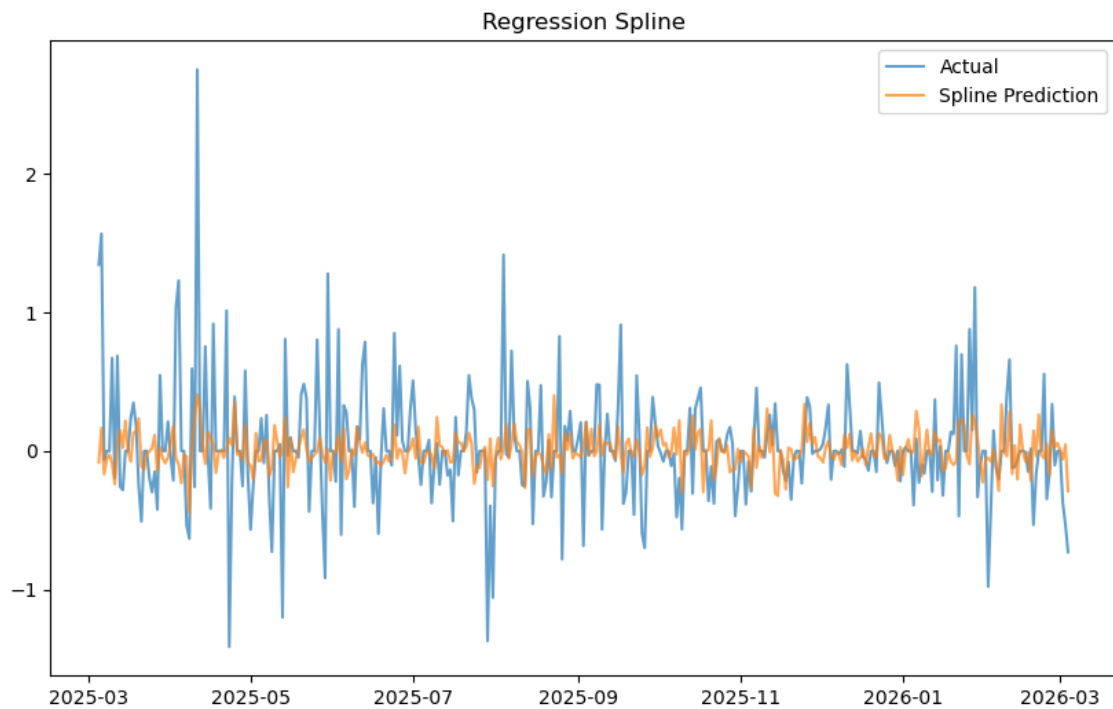


FIGURE 9 – Spline Regression (Actual vs Predicted)

4 MACHINE LEARNING MODELS

Machine learning algorithms represent a further step in predictive analysis, as they can learn complex relationships directly from the data without requiring an explicit assumption about the functional form between the predictors and the target variable. This can be particularly relevant when analysing financial markets, which are characterized by a high degree of uncertainty, structural changes and complex interactions among macroeconomic indicators, financial assets and market sentiment. In such a context, models that can identify patterns directly from the data may provide better predictive results than traditional regression approaches. Unlike the regression models considered in the previous chapters, the main objective of machine learning algorithms is not to estimate the marginal effect of each explanatory variable, but rather to obtain accurate predictions for observations that were not used during model estimation. Another important characteristic of these models is the use of regularization mechanisms and validation procedures to limit the risk of overfitting. Hyperparameters are typically selected using cross-validation or validation datasets, allowing the complexity of the model to be adjusted according to its ability to generalize to unseen observations. This chapter analyses three widely used machine learning algorithms applied to the EUR/USD forecasting problem. The first is the Random Forest, an ensemble method based on a large number of decision trees, whose predictions are aggregated to improve the overall performance of the model. The second is Extreme Gradient Boosting (XGBoost), a boosting algorithm that builds decision trees sequentially, with each new tree focusing on correcting the prediction errors made by the previous ones. Finally, the Support Vector Regression (SVR) model with a Radial Basis Function (RBF) kernel is considered. Unlike tree-based methods, SVR uses kernel functions to represent the predictors in a higher-dimensional feature space, allowing more complex nonlinear relationships to be captured while maintaining a focus on generalization performance. Overall, the objective of this chapter is to evaluate whether these machine learning algorithms can improve the prediction of daily EUR/USD returns compared with the linear and nonlinear regression models presented in the previous chapters.

4.1 Random Forest

Since the dependent variable EUR/USD is a continuous numerical variable, the Random Forest algorithm is implemented in its regression form rather than as a classification model. Random Forest Regression is an ensemble learning method that combines the predictions of multiple decision trees, each trained on different bootstrap samples of the training data and using randomly selected subsets of predictors. The final prediction is obtained by averaging the predictions of the individual trees. This approach allows the model to capture nonlinear relationships and interactions among the explanatory variables without imposing a specific functional form on the relationship between predictors and the target variable. At the same time, the use of multiple trees generally reduces the variance associated with individual decision trees and provides a more robust predictive framework. In the context of financial data, this represents a potentially useful advantage, as the relationship between exchange rates and the set of macro-financial and cryptocurrency variables considered in the analysis may not be adequately described by a linear specification. This model was evaluated on the previously defined test set, which was kept completely separate from the model estimation and tuning phases. The resulting performance measures are an MSE of 0,1568, an RMSE of 0,3960, an MAE of 0,2506, and an R^2 of 0,0585. These results indicate a noticeable improvement compared with the previously estimated linear models. In particular, the RMSE is lower than that obtained by the standard Linear Regression, Ridge and Lasso. The same can be observed for the MAE, which decreases from 0,2607 for the linear specification to 0,2506 for the Random Forest. The reduction in both error measures suggests that the tree-based model is able to capture part of the variation in the EUR/USD target that is not adequately represented by the linear specifications. Even though the explanatory power remains relatively limited in absolute terms, this result is consistent with the characteristics of daily financial returns, which are generally subject to substantial short-term fluctuations and are difficult to predict using a limited set of observable financial variables. Therefore, the relatively low R^2 should not necessarily be interpreted as evidence of model failure. Rather, it indicates that a considerable proportion of the variation in the EUR/USD target remains unexplained, even when nonlinear relationships are taken into account. At the same time, the improvement over the linear benchmarks suggests that the additional flexibility of the Random Forest is useful for the forecasting task considered in this analysis. As shown in

Figure 10, the Random Forest predictions remain generally concentrated around zero and reproduce the central tendency of the EUR/USD return series reasonably well. However, the model has more difficulty reproducing the largest positive and negative movements observed in the actual series. The predicted series is therefore smoother than the actual one, especially during periods characterized by unusually large fluctuations. This behaviour is typical of an ensemble of regression trees, since averaging the predictions of the individual trees tends to reduce the presence of extreme fitted values. As a result, while the model is able to improve the overall error measures, it does not fully capture the magnitude of exceptional market movements.

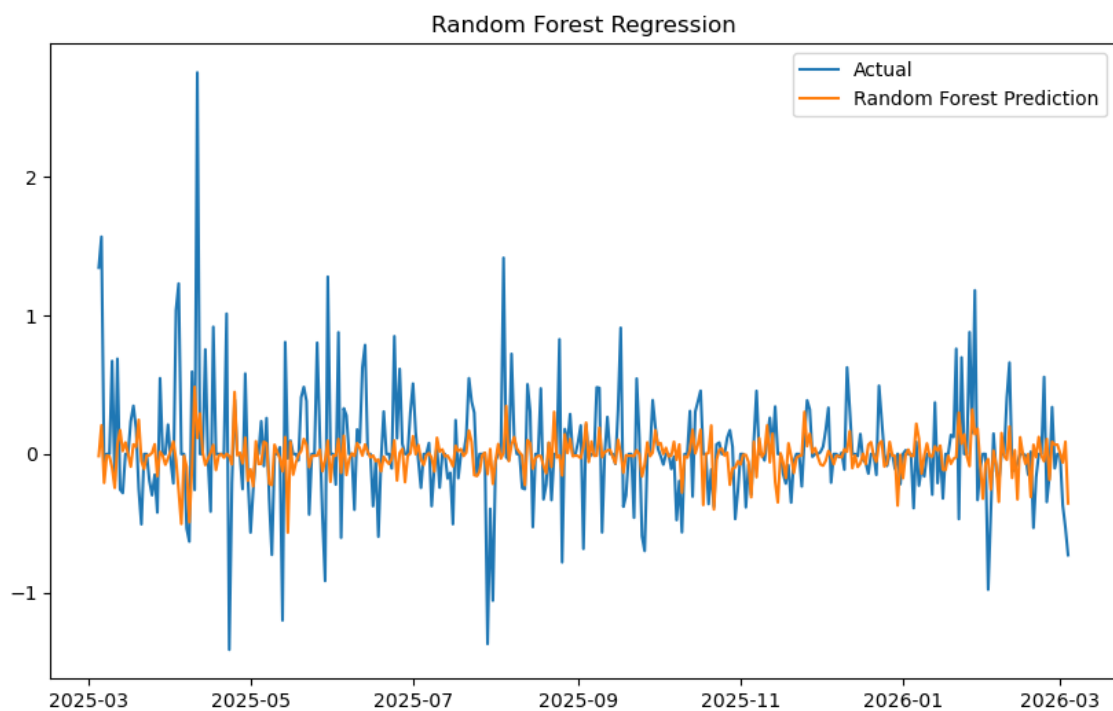


FIGURE 10 – Random Forest Regression (Actual vs Predicted)

An additional advantage of the Random Forest approach is that it provides an estimate of the relative importance of the explanatory variables. The feature-importance results, shown in Figure 11, indicate that predictive importance is concentrated among a relatively small number of variables. Gold Price is by far the most important feature, with an importance of approximately 0,27, followed by Nasdaq Price, with an importance of approximately 0,21. Treasury Yield 5Y Price ranks third, with an importance close to 0,09, while Eurostoxx50 Price and Theta Price account for approximately 0,06 and 0,05, respectively. These variables therefore play a substantially larger role in the Random Forest predictions than most of the remaining explanatory variables. The

importance assigned to Gold Price is particularly notable, it suggests that information contained in the gold market is frequently used by the individual trees when determining the predicted EUR/USD value. Similarly, the high importance of Nasdaq Price and Eurostoxx50 Price indicates that equity-market variables contribute relevant information to the prediction of the exchange rate. However, the feature-importance measure should be interpreted as an indicator of the contribution of each variable to the predictive structure of the Random Forest rather than as a causal or statistical significance measure. In particular, unlike the coefficients estimated in the OLS model, feature importance does not provide information about the direction of the relationship between a predictor and the target. The remaining variables display considerably lower importance values. Among them, Ethereum Price, GBP/USD, RUB/USD, USD Coin Price, Bitcoin Price, EUR/USD, BRL/USD, NZD/USD, CHF/USD, CNY/USD, Dai Price, and Tether Price contribute to the model to a lesser extent. This does not imply that these variables have no predictive information, but rather that, within the ensemble structure and the specific sample considered, they are used less frequently or contribute less to reducing prediction error than the variables at the top of the importance ranking.

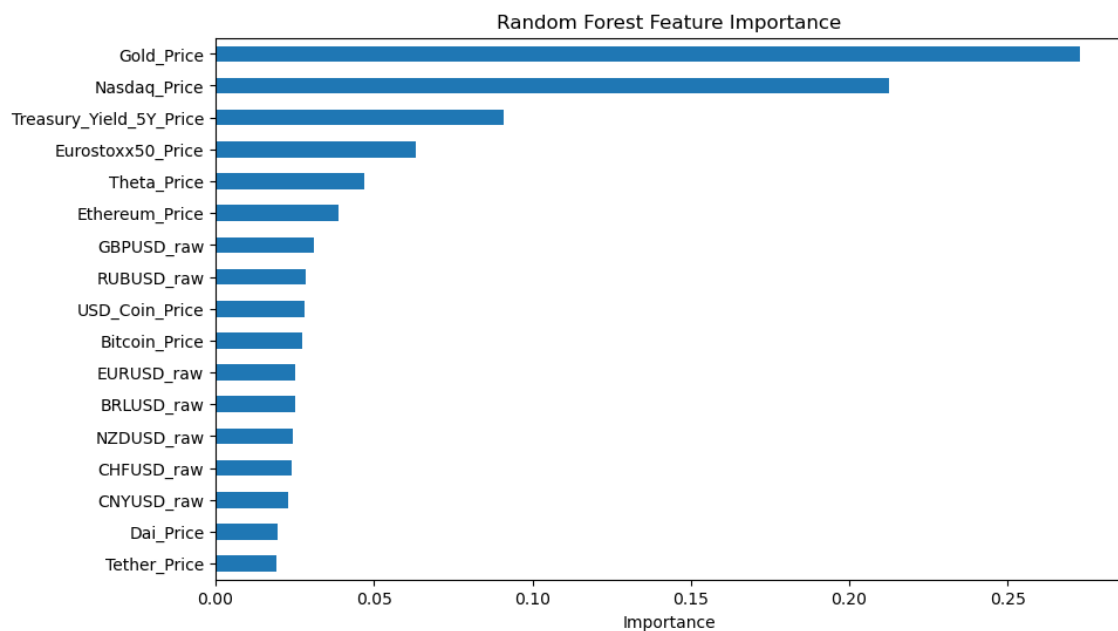


FIGURE 11 – Random Forest Feature Importance

4.2 XGBoost Model

Following the Random Forest model, the analysis proceeds with Extreme Gradient Boosting (XGBoost), a gradient boosting algorithm based on an ensemble of decision trees. XGBoost was included to investigate whether a more sophisticated tree-based learning approach could capture nonlinear relationships and interactions among the financial variables that may not be adequately represented by linear models or by the previous ensemble methods. This is particularly relevant in the present application, where the explanatory variables cover heterogeneous dimensions of the financial market, including exchange rates, equity indices, interest rates, commodities and cryptocurrencies. Unlike Random Forest, which constructs a collection of relatively independent trees and aggregates their predictions, XGBoost follows a sequential boosting strategy. Each new tree is trained to improve the errors produced by the ensemble constructed in the previous iterations. The model therefore builds a sequence of weak learners whose combined contribution progressively improves the predictive function. For a regression problem, the prediction can be represented as

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i)$$

where K represents the number of trees and f_k denotes the prediction function associated with the k -th decision tree. Rather than estimating all trees independently, XGBoost adds them sequentially, with each new tree aimed at reducing the residual prediction error of the existing ensemble. The optimization can be expressed through the objective function

$$\mathcal{L}(\emptyset) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where $l(y_i, \hat{y}_i)$ measures the prediction error and $\Omega(f_k)$ represents a regularization term controlling the complexity of the individual trees. The inclusion of the regularization component is particularly important because it limits excessive model complexity and reduces the risk of overfitting. This characteristic distinguishes XGBoost from a simple sequential tree-building procedure. The algorithm combines gradient-based optimization, tree-based learning and explicit regularization, making it suitable for complex nonlinear prediction problems. The final XGBoost specification was obtained using a learning rate of 0.01, a maximum tree depth of 3, and 300 estimators. In addition,

both “subsample” and “colsample_bytree” were set equal to 0.8. This configuration combines a relatively large number of trees with a low learning rate. The low learning rate allows the model to update its predictions gradually as new trees are added, while the use of 300 estimators provides sufficient flexibility to progressively improve the fit. At the same time, limiting the maximum tree depth to three restricts the complexity of the individual trees and therefore helps reduce the risk of fitting noise in the training data. The values of 0.8 for the observation and feature subsampling parameters introduce an additional source of regularization, since each tree is estimated using only a randomly selected proportion of the observations and explanatory variables. Overall, the selected configuration represents a relatively conservative implementation of XGBoost, which is appropriate given the limited predictability typically associated with daily exchange rate returns.

The out-of-sample results indicate that the optimized XGBoost model achieves an MSE of 0,1546, an RMSE of 0,3931, and an MAE of 0,2503. The corresponding coefficient of determination is $R^2 = 0,0721$. These results indicate that the model is able to obtain a positive, although still relatively modest, degree of explanatory power on the test set. In particular, the positive R^2 suggests that the optimized XGBoost specification performs better than a benchmark based solely on the unconditional mean of the dependent variable. However, the relatively low value of R^2 also highlights the difficulty of forecasting daily EUR/USD returns using the available financial information. This result is not unexpected, considering that the preliminary time-series analysis showed generally weak short-term linear dependence in the EUR/USD return series. Daily foreign exchange returns incorporate new information very rapidly, which limits the amount of predictable variation that can be extracted from a relatively small set of contemporaneous financial indicators. The predictive performance of XGBoost is nevertheless particularly relevant when compared with the other non-linear and ensemble approaches considered in the analysis. The model produces an RMSE of approximately 0,393, meaning that the typical magnitude of the prediction error, measured in the same units as the dependent variable, is lower than that obtained by the linear regression, Ridge, and Lasso models. The MAE of 0,2503 also indicates a relatively limited average absolute prediction error. More importantly, the positive R^2 of 0,0721 represents an improvement over the initial

XGBoost specification, for which the test-set R^2 was negative. The improvement obtained after tuning therefore suggests that the choice of hyperparameters has a meaningful effect on the model's ability to generalize to unseen observations. Nevertheless, the results should not be interpreted as evidence of strong predictability. Rather, they suggest that allowing for non-linear relationships and interactions can provide some additional predictive information, although most of the variation in daily EUR/USD returns remains unexplained.

An additional advantage of XGBoost in the context of this study is the possibility of examining the relative importance of the explanatory variables used by the model. The feature-importance results, shown in Figure 12, indicate a clear ranking among the predictors. “Gold Price” represents the most important variable, with an importance of approximately 0,134, followed by “Nasdaq Price”, with an importance of approximately 0,091. “Eurostoxx50 Price” and “Treasury_Yield_5Y Price” also have relatively high importance values, of approximately 0,084 and 0,076, respectively. These variables are followed by “CHFUSD raw”, “Ethereum Price”, “RUBUSD raw”, and “Theta Price”, whose importance values are lower but still substantial. The remaining exchange-rate variables and digital assets display more similar and relatively modest levels of importance. The prominence of “Gold Price” is particularly noteworthy. In fact, gold is traditionally regarded as an important financial asset during periods of uncertainty and may therefore contain information related to changes in market sentiment and global risk conditions. Its position as the most important feature in the XGBoost model suggests that movements in gold prices provide relevant predictive information for the next-day behaviour of the EUR/USD exchange rate within the specific sample and modelling framework considered. Similarly, the relatively high importance assigned to the Nasdaq and Euro Stoxx 50 indices indicates that international equity-market conditions may contain information that is useful for forecasting EUR/USD returns. The importance of the U.S. 5-Year Treasury Yield is also economically meaningful, as movements in U.S. interest rates can affect expectations concerning monetary policy, capital flows, and the relative attractiveness of dollar-denominated assets. The feature-importance results also show that the model does not rely exclusively on traditional macro-financial variables. Ethereum, Theta, Bitcoin, stablecoins, and other digital assets are included in the model and several of them receive non-negligible importance

values. This is relevant to the broader objective of the analysis, since the explanatory variables were deliberately selected to capture not only conventional foreign-exchange and macroeconomic information but also the growing role of digital assets in international financial markets. The inclusion of these variables allows the model to investigate whether information originating from cryptocurrency markets contributes to the prediction of EUR/USD returns. The dataset was constructed precisely to combine exchange rates, equity markets, commodities, government bond yields, and digital assets in a common forecasting framework. It is important, however, to distinguish feature importance from statistical significance or causality. A high importance score does not imply that a variable has a statistically significant effect on EUR/USD returns, nor does it establish that changes in that variable cause changes in the exchange rate. Instead, the importance measure reflects the contribution of the variable to the predictive structure learned by the XGBoost algorithm. This distinction is particularly important when comparing XGBoost with the OLS models presented earlier in the analysis. In the linear framework, coefficients and their associated p-values provide information about the direction and statistical significance of conditional relationships under the assumptions of the model. XGBoost, in contrast, is primarily predictive and does not provide the same type of inferential interpretation. Consequently, the feature-importance ranking should be interpreted as evidence concerning the predictive relevance of the variables rather than their causal or statistical significance. Overall, the empirical results indicate that XGBoost provides a meaningful improvement in predictive performance relative to the initial specification and achieves a positive out-of-sample R^2 . Its ability to model non-linear relationships and interactions among heterogeneous financial variables appears to provide additional information compared with purely linear approaches. At the same time, the relatively low R^2 confirms that the forecasting problem remains challenging and that the majority of the variation in daily EUR/USD returns cannot be explained by the variables included in the model. The results therefore suggest that increasing model complexity alone is not sufficient to obtain highly accurate forecasts in the foreign exchange market. Rather, the main value of XGBoost in this study lies in its ability to explore more complex predictive structures and to identify which financial variables contribute most to the forecasting process. This provides a useful transition towards the subsequent analysis of more advanced modelling techniques, where the objective is to assess whether further

increases in model flexibility can lead to additional improvements in out-of-sample forecasting performance.

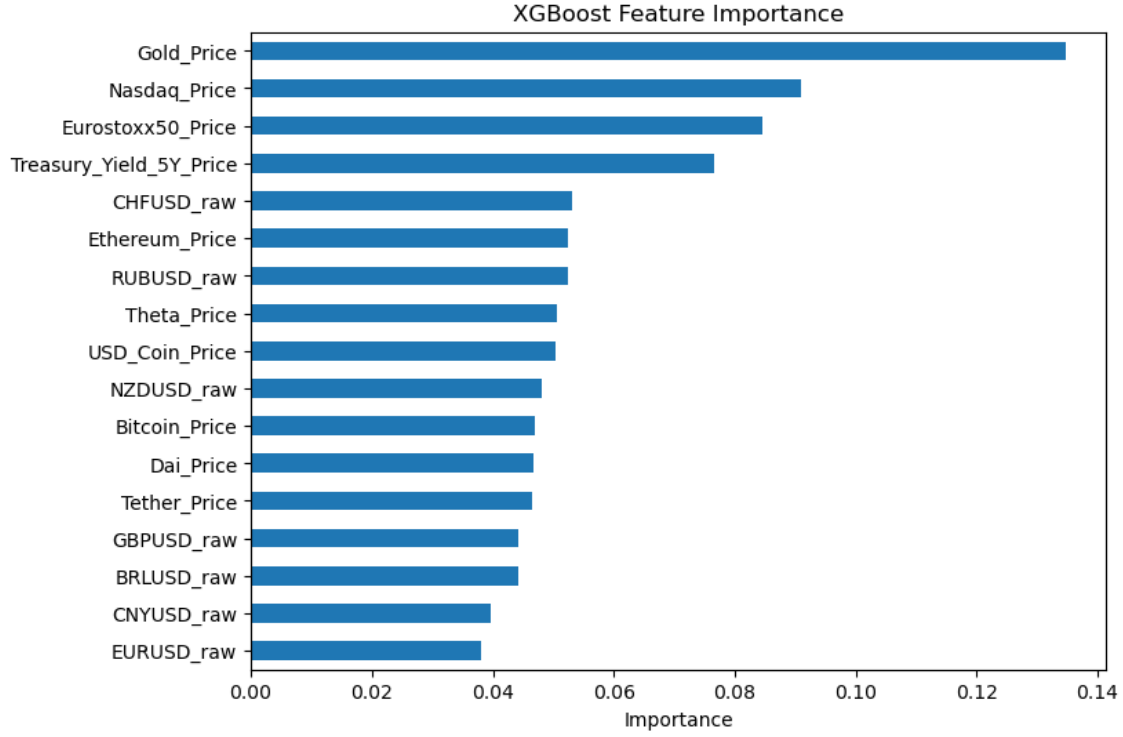


FIGURE 12 – XGBoost Feature Importance

4.3 Support Vector Machines with RBF kernel

The next machine learning model considered in this analysis is Support Vector Regression (SVR) with a Radial Basis Function (RBF) kernel. The analysis considers SVR for a specific reason: unlike tree-based methods such as Random Forest and XGBoost, which partition the feature space through recursive splits, SVR takes a fundamentally different approach. It maps the original predictors into a higher-dimensional space using a kernel function, and then fits a linear regression in that transformed space. This makes it particularly well-suited for capturing complex, nonlinear relationships that may not be easily represented by decision trees. The RBF kernel, in particular, has the following form:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

where γ controls the width of the kernel and therefore the degree of smoothness of the resulting decision function. In practice, choosing the right value of γ is crucial: if it is too

large, the model becomes overly sensitive to local fluctuations and may overfit; if it is too small, the model may become too smooth and fail to capture relevant patterns in the data. The SVR formulation also introduces two additional hyperparameters: C , which controls the trade-off between fitting the training data and keeping the model simple, and ϵ , which defines a margin of tolerance around the predicted values. Observations falling within this margin do not contribute to the loss function, which makes SVR relatively robust to outliers (a useful property when dealing with financial data that often contain extreme observations). For the tuning process, a grid search was performed over a range of candidate values for C , ϵ , and γ . The optimal configuration, selected based on the lowest validation error, was:

- $C = 1$,
- $\epsilon = 0,01$,
- $\gamma = 0,01$.

The relatively low value of C suggests that the model favours a simpler decision function, rather than trying to fit every training observation perfectly. This is consistent with the general finding throughout this analysis that daily exchange rate returns are difficult to predict, and that aggressive model complexity tends to hurt rather than help out-of-sample performance. The small value of ϵ indicates that the model maintains a relatively narrow margin of tolerance, which means that more observations are allowed to influence the fit. The model achieves an MSE of 0,1608, an RMSE of 0,4010, and an MAE of 0,2514. While the corresponding coefficient of determination is $R^2 = 0,0347$. These values place the SVR model somewhere in the middle of the performance ranking among the models considered so far. The RMSE is slightly higher than that of XGBoost (0,393) and Spline Regression (0,389), but lower than the linear models and Random Forest. The same pattern can be observed for the MAE, which is comparable to XGBoost (0,250) and better than the linear specifications. The R^2 is positive, which is reassuring: the model does at least as well as a simple mean forecast. However, it is considerably lower than the one achieved by Spline Regression (0,090) and XGBoost (0,072). This suggests that, within the set of nonlinear approaches, SVR with an RBF kernel is not the most effective at capturing the relationships that drive EUR/USD movements. Figure 13 shows the actual EUR/USD returns against the SVR predictions on the test set. The plot reveals a pattern

that has become familiar throughout this empirical analysis: the predicted series is considerably smoother than the actual one. The SVR forecasts remain concentrated around zero and follow the general tendency of the observed returns, but they consistently fail to reproduce the magnitude of the larger positive and negative movements. During periods of relative calm, the model tracks the data reasonably well; during episodes of heightened volatility, however, the gap between the predicted and actual values widens noticeably. Moreover, the ϵ -insensitive loss function means that the model is not penalised for small prediction errors, which encourages a flat and stable decision surface. While this makes SVR robust to outliers, it also means that the model tends to under-predict extreme movements; this limitation is particularly relevant in financial applications, where large price swings are often the most important events to capture. Another point worth considering is that SVR, unlike tree-based methods, does not provide a straightforward measure of feature importance. This represents one of the main trade-offs associated with the use of this model. While the kernel approach allows SVR to capture complex nonlinear relationships, the resulting model is considerably less interpretable than a linear regression and, to some extent, than a Random Forest. In the context of this study, this limitation is acceptable because the main objective of the analysis is predictive performance rather than the direct interpretation of individual coefficients or predictors. Comparing the SVR results with those obtained from the other models, a few aspects are particularly relevant. The improvement over the linear benchmarks is relatively modest. The R^2 of 0,034 is only slightly higher than the values obtained by Lasso (0,032) and Ridge (0,031), while the RMSE is also relatively close to those models. This suggests that, for the dataset considered in this study, the additional flexibility introduced by the RBF kernel does not lead to a substantial improvement in predictive accuracy. At the same time, the difference between SVR and the best-performing models, namely Spline and XGBoost, is more evident. Both of these models achieve lower prediction errors and higher R^2 values, suggesting that they are better able to capture the structure present in the data. One possible explanation for the relatively modest performance of SVR is related to the characteristics of the feature space. The RBF kernel implicitly maps the predictors into a higher-dimensional representation, which can be useful when the relationship between the inputs and the target is strongly nonlinear. However, when the underlying predictive signal is weak, as is often the case with daily

exchange rate returns, this additional flexibility may also make the model more sensitive to noise. In such a situation, the model may capture relationships that are present in the training data but do not generalize well to the test set. This interpretation is consistent with the relatively low R^2 obtained by the model and with the value of C selected during the validation procedure. Another relevant aspect is the sensitivity of SVR to the scaling of the input variables and to the choice of the kernel parameters. This is why the predictors were standardized before estimating the model. Despite the grid search performed for the hyperparameter selection, it is possible that a different range of values or a different kernel specification could have produced better results. However, the range considered in the analysis was reasonably broad, and the selected hyperparameter values appear consistent with the characteristics of the dataset. Overall, the SVR model with an RBF kernel provides a useful comparison within the broader machine learning analysis. It achieves positive out-of-sample performance and slightly improves upon the linear benchmarks, but it does not outperform the best nonlinear models considered in this study. The results suggest that kernel methods provide a flexible and theoretically sound way of introducing nonlinear relationships, but they are not necessarily the most effective approach for this particular forecasting problem. In this case, tree-based ensembles and spline regression achieve better predictive results. This comparison also highlights that the performance of a given modelling approach depends on the characteristics of the data and that greater model complexity does not necessarily translate into better out-of-sample predictions.

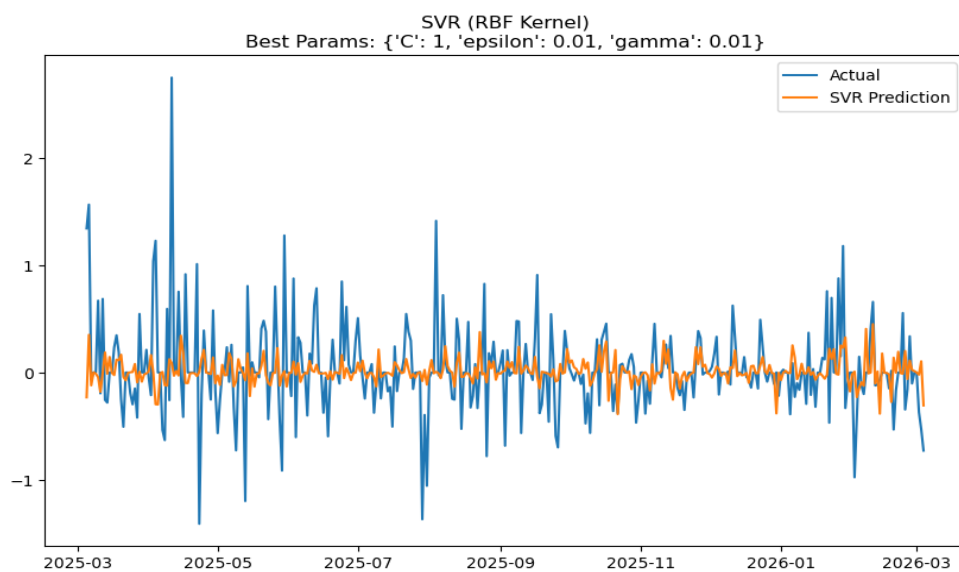


FIGURE 13 – SVM with RBF Plot (Actual vs Predicted)

5 DEEP LEARNING MODELS

The empirical analysis so far has followed a progression from simple linear models to regularized specifications, nonlinear parametric approaches, and finally tree-based and kernel-based machine learning algorithms. Each of these approaches has provided useful results, but some common patterns can also be identified. The linear models, despite their high degree of interpretability, achieved only modest out-of-sample performance, with R^2 values only slightly above zero. The nonlinear parametric models, particularly spline regression, produced a noticeable improvement, suggesting that the relationship between the financial predictors and the EUR/USD return may not be adequately described by a purely linear specification. The machine learning models, especially XGBoost, provided a further improvement by capturing nonlinearities and interactions that were not fully represented by the previous models. Moreover, daily exchange rate returns are generally difficult to forecast, and the weak linear autocorrelation observed in the ACF analysis suggests that the EUR/USD series contains only limited information from its own past. However, the absence of strong linear autocorrelation does not necessarily imply that the series is completely independent over time. More complex temporal dependencies may still be present, including patterns that develop over several days or interactions between variables that are difficult to capture using the models considered so far. Deep learning provides a further approach to this problem. Unlike the models considered in the previous sections, deep neural networks can learn complex representations of the data through multiple layers of nonlinear transformations. They do not require a specific functional form to be defined in advance and can, in principle, identify relevant patterns directly from the input data. This flexibility represents an important advantage, but it also increases the risk of overfitting. A sufficiently complex neural network can approximate very flexible relationships, but its performance may deteriorate when the model starts fitting noise rather than patterns that generalize to unseen observations. For this reason, appropriate regularization and validation procedures are particularly important when applying deep learning to financial time series. For this analysis, three deep learning architectures were selected, each representing a different approach to the forecasting problem. The first is the Multi-Layer Perceptron (MLP), which represents the simplest form of feedforward neural network considered in this

analysis, and is included as a baseline within the deep learning framework. The second model is the Long Short-Term Memory (LSTM) network, that is a type of recurrent neural network specifically designed to work with sequential data. The third and most advanced architecture considered is the Temporal Fusion Transformer (TFT), that combines recurrent components with attention mechanisms and is designed to handle temporal dependencies in forecasting problems. The use of deep learning models also introduces several practical challenges that are less relevant for the regression and machine learning models considered previously. The choice of architecture, number of layers, activation functions, optimization algorithm, learning rate, batch size and number of epochs can all affect the resulting performance. Overfitting is another important concern, particularly given the relatively limited number of daily observations available for the analysis. To address these issues, several strategies were adopted, including early stopping to avoid unnecessary training, dropout as a regularization technique, and a separate validation set for hyperparameter tuning. Particular attention was also paid to preventing information from the test set from entering the training process, which is essential when evaluating forecasting models on time-series data.

The main objective of this chapter is therefore to assess whether deep learning models, with their greater flexibility and ability to learn complex patterns, can outperform the best-performing models considered in the previous chapters. If the deep learning models provide substantially better results, this would suggest that the data contain additional structure that was not captured by the simpler specifications. Conversely, if their performance remains limited, this would provide further evidence that the high degree of noise and unpredictability in daily exchange rate returns represents a more fundamental limitation, rather than simply a consequence of model choice. The following sections present the results obtained from the three deep learning architectures. Their predictive performance is then compared both across the three deep learning models and with the best-performing models from the previous chapters.

5.1 Multi-Layer Perceptron Model

The first deep learning model considered is the Multi-Layer Perceptron (MLP), which represents the simplest form of feedforward neural network. It is included as a baseline within the deep learning framework. Unlike the LSTM and TFT models considered later, the MLP does not explicitly incorporate temporal structure. Instead, it learns a nonlinear mapping between the input features and the target variable without directly modelling the sequential order of the observations. In this sense, it is conceptually closer to the nonlinear regression and tree-based models examined in the previous sections, although it uses a different functional form based on successive affine transformations and nonlinear activation functions. The architecture used in the analysis is relatively simple. The network consists of an input layer corresponding to the explanatory variables, followed by two hidden layers, each containing 64 neurons with ReLU activation functions. A relatively shallow architecture and a moderate number of units were selected for two main reasons. First, the dataset is not particularly large, with approximately 1,500 training observations, so a deeper network could have increased the risk of overfitting. Second, daily exchange rate returns are generally characterized by a relatively low signal-to-noise ratio. Therefore, there is limited reason to expect that a highly complex network would necessarily capture meaningful patterns that could not be captured by a simpler architecture. Dropout with a rate of 0,2 was applied between the hidden layers. Dropout is a regularization technique that randomly sets a fraction of the activations to zero during training. This forces the network to rely less on individual neurons and can therefore help it learn more robust representations while reducing the risk of overfitting. A dropout rate of 0,2 provides a relatively moderate degree of regularization, allowing the network to retain sufficient capacity while still limiting the possibility of excessive fitting to the training observations. The network was trained using the Adam optimizer, with a learning rate of 0,001 and a batch size of 32. These are standard choices for this type of model and were not substantially modified during the tuning process. The loss function was the Mean Squared Error (MSE), consistently with the evaluation metrics used throughout the analysis. The validation loss was also monitored during training, and early stopping with a patience of 10 epochs was implemented. This means that training was stopped when the validation loss did not improve for 10 consecutive epochs. The purpose of this procedure was to limit overfitting and avoid continuing the training process once the model had

stopped improving on unseen validation observations. Training stopped after 54 epochs, while the best validation loss was approximately 0,1256. Figure 14 reports the learning curves for both the training and validation sets. The training loss decreases steadily during the first 20–30 epochs and then continues to decline at a slower rate. The validation loss follows a different pattern: it decreases during the initial stages of training, reaches its minimum around epoch 20–25, and then gradually increases despite the continued reduction in training loss. This divergence between training and validation loss is a typical indication of overfitting, since the network continues to improve its fit to the training data without achieving corresponding improvements on the validation set. Although early stopping was applied, the training process continued until epoch 54, well after the validation loss had started to increase. This suggests that the selected patience value may have allowed the model to continue training beyond the point of optimal generalization.

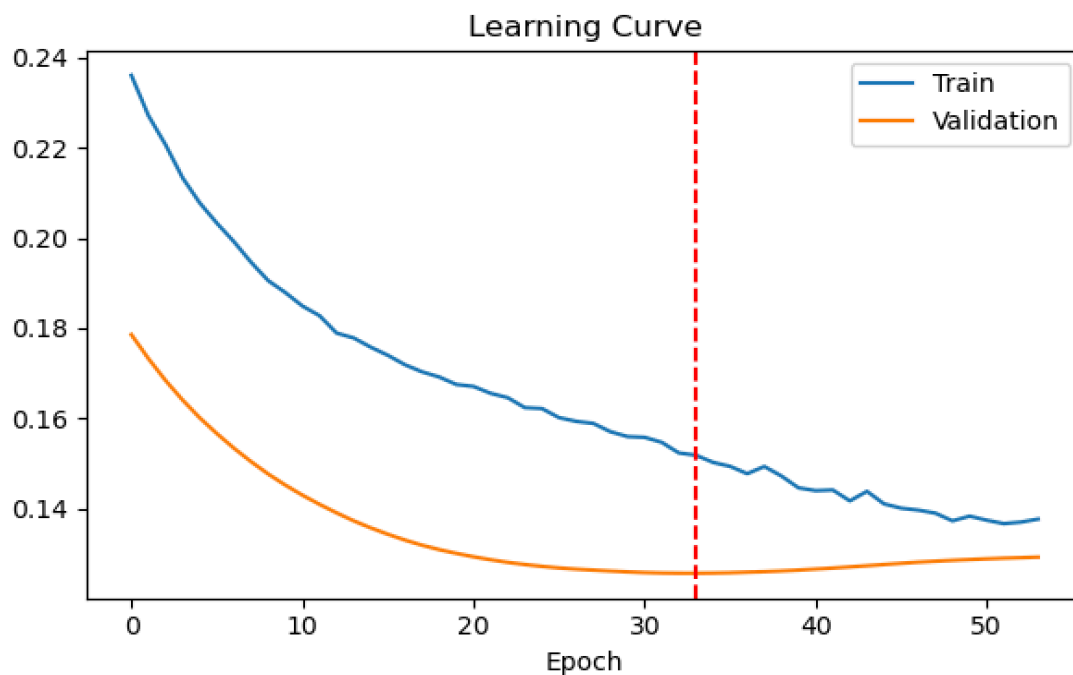


FIGURE 14 – MLP Learning Curve

The out-of-sample performance on the test set provides further evidence of the model's limited generalization ability. The MLP achieves an MSE of 0,1690, an RMSE of 0,4111, and an MAE of 0,2628. The resulting R^2 is negative, equal to $-0,0144$. This means that the model performs worse than a simple benchmark that always predicts the mean of the dependent variable. In other words, the information contained in the predictors does not translate into sufficiently accurate out-of-sample predictions for this particular MLP

specification. This result is considerably weaker than expected when compared with the other models considered in the analysis. The MLP produces an R^2 substantially lower than that obtained by OLS (0,0218) and Lasso (0,0323), and it also performs poorly compared with the other nonlinear specifications. Polynomial Regression is the only other model with a negative out-of-sample R^2 , although its value of $-0,5875$ reflects a considerably different degree of deterioration. Among the models considered so far, the MLP is therefore one of the few specifications that fails to achieve a positive out-of-sample R^2 . Figure 15 shows the MLP predictions together with the actual EUR/USD returns on the test set. The visual comparison is also informative. The predicted series is smoother than those produced by several of the other models and remains tightly concentrated around zero throughout most of the test period. The actual series, in contrast, displays considerably more variability, including several pronounced positive and negative movements. The MLP is unable to reproduce most of these larger fluctuations. Its predictions therefore remain relatively close to the mean, although this is not sufficient to obtain a positive R^2 .

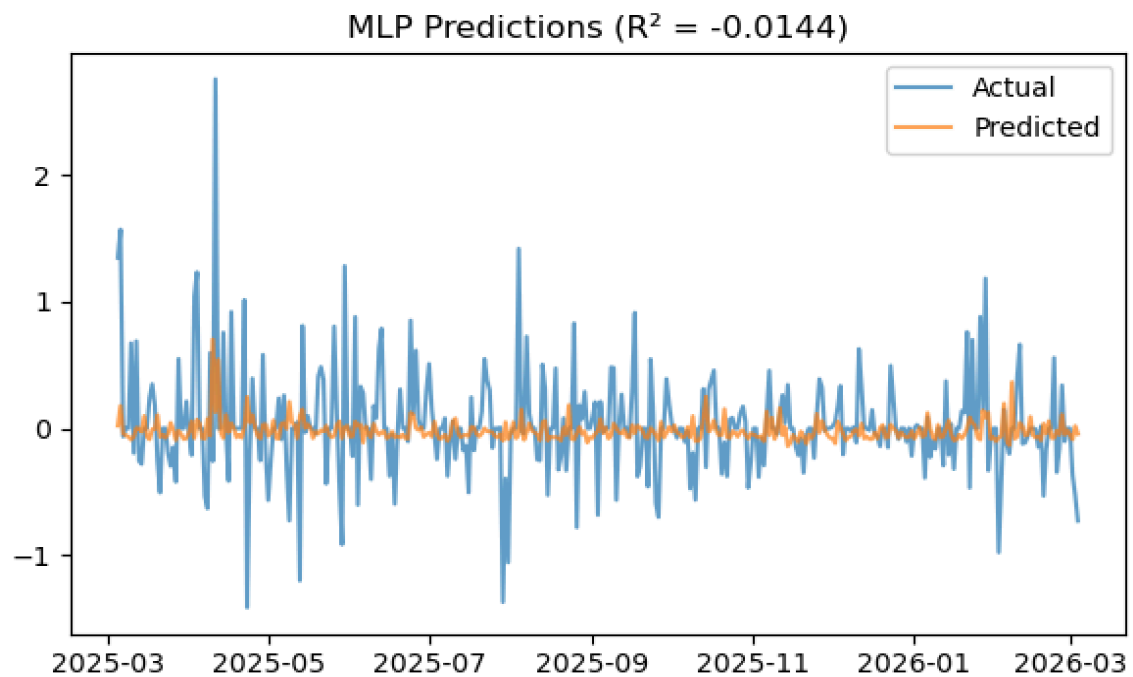


FIGURE 15 – MLP Plot (Actual vs Predicted)

Several factors may help explain this result. The first is overfitting, despite the use of early stopping and dropout. The learning curves show a clear divergence between training and validation loss, suggesting that the model continues to learn patterns specific to the training sample that do not generalize well to unseen observations. The fact that the best validation loss occurs around epochs 20–25, while training continues until epoch 54, also suggests that the selected patience value may have been relatively high. A more restrictive early stopping criterion, for example with a patience of 5 epochs, could potentially have resulted in a model with better generalization performance. A second possible explanation concerns the architecture of the MLP itself. Feedforward networks are designed to learn mappings between inputs and outputs, but they do not explicitly model temporal dependencies. Such dependencies can only be represented indirectly through the features provided to the network. In the present analysis, the predictors are shifted by one period, so the model does incorporate lagged information. However, if relevant information is distributed across several time steps, or if interactions between past values of different variables are important, the MLP may not be able to capture this structure effectively. This represents an important difference compared with recurrent architectures such as the LSTM, which are specifically designed to process sequential information. A third explanation is related to the amount of predictive information contained in the data. The linear models achieved R^2 values of only around 2–3%, while the best-performing models, particularly spline regression and XGBoost, achieved values in the range of approximately 7–9%. If the underlying predictability of daily EUR/USD returns is intrinsically limited, even a well-tuned neural network will face difficulties in producing highly accurate forecasts. The negative obtained by the MLP may therefore be interpreted as an extreme case in which the additional flexibility of the model allows it to fit noise rather than patterns that are stable out of sample.

It is also possible that the MLP was not tuned as extensively as some of the other models. The architecture used in this analysis is relatively simple and standard hyperparameter values were adopted, while no extensive grid search was performed for the MLP comparable to the procedures used for SVR and XGBoost. A different combination of the number of layers, number of units, learning rate or dropout rate could potentially have produced better results. However, given the clear overfitting pattern observed in the learning curves and the fact that the validation loss began to increase relatively early

during training, more extensive tuning may not have fundamentally changed the overall performance of the model.

The MLP results should therefore be interpreted within the broader context of the analysis. The negative R^2 indicates that this particular feedforward architecture is not well suited to the forecasting problem under consideration. However, this result does not necessarily imply that deep learning models are unsuitable for the EUR/USD forecasting task as a whole. The MLP result also highlights an important point regarding the use of deep learning in financial forecasting. Neural networks can provide considerable flexibility, but this flexibility does not automatically lead to better predictions. Their performance depends on factors such as the amount of available data, the degree of regularization, the choice of hyperparameters and, importantly, the suitability of the architecture for the structure of the problem. In the present case, the MLP does not explicitly account for the sequential nature of the data, which may limit its ability to capture temporal patterns relevant for EUR/USD returns. Its poor performance should therefore be interpreted as a limitation of this particular architecture in the present setting rather than as evidence against deep learning more generally. Overall, the MLP does not achieve competitive out-of-sample performance. It produces a negative R^2 , with prediction errors higher than those obtained by the linear models and the better-performing tree-based and nonlinear specifications. The learning curves indicate signs of overfitting, while the predicted values are too smooth to reproduce the variability observed in the actual EUR/USD returns. Nevertheless, the model serves an important purpose in the analysis by providing a baseline for the deep learning section and showing that a simple feedforward neural network is not sufficient for the forecasting task considered here.

5.2 Long Short-Term Memory Model

The second deep learning model considered is the Long Short-Term Memory (LSTM) network. Unlike the MLP, which does not explicitly account for the sequential structure of the observations, the LSTM is specifically designed to process sequential data. It belongs to the family of recurrent neural networks (RNNs) and uses a gating mechanism that allows the model to retain relevant information over multiple time steps, while also controlling which information should be forgotten or passed forward. This makes the LSTM particularly suitable for time-series forecasting, as it can potentially capture dependencies that extend beyond the immediate past. The inclusion of the LSTM in the analysis is motivated by the results obtained from the ACF and PACF analysis presented in Section 1.4. These results showed that the linear autocorrelation in the EUR/USD return series is relatively weak, with most lags remaining within the confidence bands and only a few exceeding them by a relatively small amount. However, weak linear autocorrelation does not necessarily exclude the presence of more complex nonlinear temporal dependencies. Exchange rate movements may depend on patterns that develop over several days or on interactions between past values of different variables that cannot be adequately captured by a linear model. The LSTM is particularly suited to this type of structure and therefore represents a relevant model for the forecasting problem considered in this study. Before estimating the final model, a systematic hyperparameter search was carried out in order to identify a configuration with better expected generalization performance. Four main hyperparameters were considered: the sequence length, corresponding to the number of past observations used as input; the number of hidden units; the number of recurrent layers; and the learning rate. The search considered sequence lengths of 5, 10, and 15 days, hidden units of 32 and 64, one or two recurrent layers, and learning rates of 0,001 and 0,005. For each combination, the model was trained on the training set and evaluated on the validation set, with the validation loss used as the criterion for selecting the best configuration. The results of the hyperparameter search are reported in the output below. The configuration associated with the lowest validation loss was the following:

- Sequence length (seq_len): 10,
- Hidden units: 64,

- Number of layers: 2,
- Learning rate: 0,005.

The choice of a sequence length of 10 is particularly interesting. It means that, within the LSTM framework, the model uses approximately two weeks of previous trading observations when predicting the following day's return. This is longer than the lags that appeared to be relevant in the ACF analysis, which were mainly concentrated in the first few days. One possible interpretation is that the LSTM is able to capture nonlinear patterns extending over several days that are not visible through a linear autocorrelation analysis. At the same time, it is also possible that the additional historical information contributes to fitting noise rather than stable predictive patterns. The relatively weak performance obtained on the test set is consistent with this second possibility. The selection of 64 hidden units and two recurrent layers indicates that the best validation performance was obtained with a relatively flexible architecture. A single layer with 32 units may have provided insufficient capacity, while the two-layer configuration allowed the network to learn more complex representations from the sequential input. The relatively high learning rate of 0,005 is also noteworthy, as it suggests that the selected configuration was able to converge with relatively large optimization steps without requiring a smaller learning rate.

Once the optimal hyperparameters had been identified, the final LSTM was trained on the full training set using early stopping to limit overfitting. The training process stopped after only 16 epochs, which is considerably shorter than the 54 epochs required by the MLP. This relatively early stopping suggests that the validation loss started to deteriorate quickly during training. One possible explanation is that the LSTM, given its greater complexity, began to overfit the training observations relatively early. In this case, early stopping acted as a regularization mechanism by preventing the model from continuing to adapt to patterns that were not improving its performance on the validation data. The out-of-sample performance on the test set is considerably weaker than that obtained by the other models considered so far. The LSTM achieves an MSE of 0,2130, an RMSE of 0,4615, a MAE of 0,2607, and an R^2 of -0,3624. This represents the worst test-set performance among the models considered up to this point. The substantially negative R^2 indicates that the LSTM performs considerably worse than a simple

benchmark based on predicting the mean of the dependent variable. Therefore, the additional temporal flexibility provided by the recurrent architecture does not translate into improved predictive accuracy for the EUR/USD forecasting problem considered here. Figure 16 reports the LSTM predictions together with the actual EUR/USD returns on the test set. The visual comparison provides further evidence of the model's limited predictive ability. While the actual series shows considerable variability, including frequent positive and negative movements, the predicted values remain almost completely flat. They fluctuate only slightly around a relatively constant level and therefore reproduce very little of the variation observed in the actual series. The model appears to rely heavily on a value close to zero when generating its forecasts. However, this behaviour is not sufficient to achieve a positive R^2 , since the predicted values remain almost constant while the actual returns display considerably greater variability.

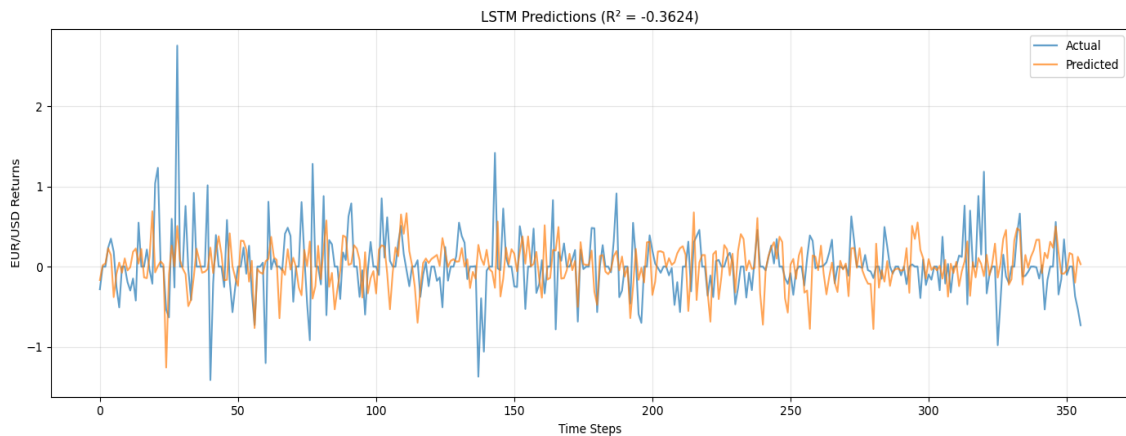


FIGURE 16 – LSTM Plot (Actual vs Predicted)

Several factors may help explain this result. The first concerns the complexity of the LSTM relative to the amount of available data. The training sample contains approximately 1500 observations, while the selected model includes two recurrent layers and 64 hidden units. This provides the network with considerably more flexibility than the MLP and may increase the risk of overfitting. The fact that early stopping occurred after only 16 epochs suggests that the model began to lose generalization performance relatively quickly. Moreover, even the configuration that achieved the lowest validation loss may have captured patterns that were specific to the training and validation periods but did not persist in the independent test sample. The best validation loss was 0,1575 and

was obtained with a sequence length of 10 and two recurrent layers, indicating that the selected model was able to fit the available data reasonably well but that this performance did not translate into comparable test-set results. A second possible explanation is related to the structure of the data itself. LSTMs are designed to identify sequential patterns and potentially long-range dependencies. However, if the EUR/USD return series contains relatively little temporal structure, as suggested by the ACF analysis, there may be limited information for the model to exploit. In such a situation, the flexibility of the LSTM can become a disadvantage, since the model may fit noise in the training sample rather than stable relationships that remain present in the test period. The negative test-set R^2 is consistent with this interpretation, as it indicates that the patterns learned by the model did not generalize effectively to unseen observations. A third possible explanation concerns the hyperparameter selection procedure. The optimal configuration was selected according to validation loss, and therefore the validation set played an important role in determining the final architecture. Although the test set remained separate from this process, repeated selection based on the validation sample can still lead to a configuration that performs particularly well for the specific characteristics and noise patterns of that period. In this case, the sequence length of 10 and the two-layer architecture may have been well suited to the validation observations but less appropriate for the subsequent test period. This highlights the importance of maintaining a completely separate test set for the final evaluation of forecasting performance.

The LSTM results also differ considerably from those obtained with the MLP. Although the MLP produced a negative R^2 , its predictions still showed a limited ability to reproduce the general tendency of the series. The LSTM predictions, in contrast, are almost constant over the test period. This suggests that the additional sequential structure introduced by the recurrent architecture did not improve the model's ability to exploit the available information. Instead, the greater complexity of the LSTM may have limited its generalization performance in the present setting. These results also provide some insight into the broader question of EUR/USD return predictability. One possible interpretation is that the available data contain insufficient temporal structure to justify the additional complexity of a recurrent model. The weak autocorrelation identified in the ACF analysis, together with the relatively limited predictive power of the linear models, points in this direction. The LSTM results provide further evidence that explicitly modelling sequential

dependencies does not necessarily improve forecasting performance when the underlying temporal signal is weak. Another possible interpretation is related to the characteristics of daily exchange rate returns. If the series behaves approximately as a random walk, predictable movements may be very limited. In the case of returns, a simple forecast close to zero can represent a strong benchmark. The flat predictions generated by the LSTM can therefore be interpreted as an extreme case in which the model converges towards a nearly constant forecast. However, the model performs worse than the mean benchmark, as indicated by its substantially negative R^2 .

From a methodological perspective, the LSTM results highlight some of the limitations associated with applying deep learning to financial forecasting. The model was tuned through a hyperparameter search, trained using early stopping, and evaluated on a separate test set. Despite these procedures, its out-of-sample performance was considerably worse than that of the simpler models. This suggests that, when the signal-to-noise ratio is low and the underlying temporal structure is limited, increasing model complexity does not necessarily result in better forecasts. In some cases, it can instead lead to poorer generalization performance. Overall, the LSTM represents the worst performing specification among the models considered so far. Its R^2 of approximately $-0,362$ is substantially lower than that of the MLP and the linear benchmarks, while the RMSE and MSE are also comparatively high. The almost flat predictions and the early stopping after 16 epochs are consistent with the limited ability of the model to extract stable predictive patterns from the available data. Although LSTMs can be highly effective in time-series forecasting applications, the results obtained here suggest that the additional sequential modelling capacity does not provide an advantage for the EUR/USD forecasting problem considered in this study.

5.3 Temporal Fusion Transformer Model

5.3.1 Structure and features

The third and final deep learning model considered in this analysis is the Temporal Fusion Transformer (TFT). Unlike the MLP and LSTM, which are relatively established architectures in the time-series forecasting literature, the TFT represents a more recent approach that combines several mechanisms within a single forecasting framework. It was included in the analysis for two main reasons. First, it is specifically designed for interpretable time-series forecasting. Second, its architecture provides some features that can potentially address the limitations observed in the MLP and LSTM models. The MLP does not explicitly account for the temporal structure of the observations, while the LSTM is specifically designed for sequential data but produced considerably weaker results in the present application. The TFT adopts a different approach by combining recurrent layers with self-attention mechanisms. Rather than processing the sequence exclusively in a sequential manner, as in a conventional recurrent architecture, the attention mechanism allows the model to assign different levels of importance to different time steps. This makes it possible to consider both short-term and longer-term dependencies and to determine dynamically which parts of the input sequence are more relevant for a given prediction. One of the main features of the TFT is its ability to distinguish between static and dynamic covariates. Static covariates are variables whose values do not change over time. In the context of this analysis, these could include macroeconomic regime indicators or other time-invariant characteristics, if included in the model. Dynamic covariates, in contrast, vary over time and include both past values of the target variable, represented by EUR/USD returns, and other time-varying predictors such as stock indices, commodity prices, and cryptocurrency returns. Treating these two types of information separately allows the model to account for their different roles in the forecasting process and can potentially improve the representation of the relationships between the predictors and the target variable. Another important characteristic of the TFT is its focus on interpretability. For deep learning models, it is generally difficult to determine which aspects of the input data have contributed most to a specific forecast. The TFT addresses this issue through its attention mechanism and variable-selection components. The attention weights provide information on the relative importance

assigned to different time steps when generating a prediction. These weights can be visualized and examined in order to obtain a better understanding of which parts of the historical sequence contribute most to the model's forecast. This feature is particularly relevant in a financial application, where the interpretation of the factors underlying a prediction can be useful alongside the prediction itself. The architecture of the TFT consists of several distinct components. First, a sequence-to-sequence layer processes the input sequence and encodes information from the previous time steps into hidden representations. This is followed by a multi-head attention mechanism, which allows the model to focus on different portions of the input sequence simultaneously. The attention mechanism produces a weighted combination of the encoded representations, with the weights reflecting the relative relevance of the different time steps for the prediction being generated. This mechanism is conceptually related to the attention mechanism used in Transformer architectures for natural language processing, but is adapted to the specific characteristics of time-series forecasting. The TFT also incorporates variable-selection mechanisms that allow the model to learn which predictors are more relevant for the forecasting task. This is particularly useful in the present analysis because the dataset contains a relatively large number of explanatory variables belonging to different financial and asset classes. Instead of assigning the same importance to all predictors, the model can learn to focus on those that provide greater predictive information. This can potentially improve generalization performance while also providing additional information about which financial variables contribute most to the EUR/USD forecasting task. Moreover, from a practical perspective, implementing the TFT is considerably more complex than implementing the MLP or LSTM. The architecture contains a larger number of components and hyperparameters, and the training procedure therefore requires greater attention to model configuration and computational resources. The PyTorch Forecasting library was used to implement the TFT, providing a practical framework for constructing and training the model. Even with this support, however, the TFT required more computational resources and a greater tuning effort than the other deep learning architectures considered in this study. The main question is therefore whether the additional complexity of this architecture translates into an improvement in out-of-sample predictive performance for the EUR/USD forecasting problem.

5.3.2 Empirical Implementation

The TFT was implemented using the “PyTorch Forecasting” library, which provides a high-level framework for constructing and training Temporal Fusion Transformer models. This library was selected because it includes the main components of the TFT architecture, such as the variable selection mechanisms, interpretable multi-head attention, and quantile forecasting. It also integrates with “PyTorch Lightning”, which simplifies the training procedure and provides built-in functionalities for early stopping, model checkpointing, and training monitoring. According to the model output, the resulting architecture consists of 20 distinct layers and approximately 33,7K trainable parameters. This is a relatively small number of parameters compared with some TFT implementations used in the literature, where the number of parameters can reach several million. The relatively compact architecture is consistent with the size of the dataset, which contains approximately 1500 training observations, and with the decision to adopt a relatively conservative model specification in order to limit the risk of overfitting. The main components of the architecture are the following:

- Input embeddings (1,2K parameters): these transform the original input features into learned representations that can subsequently be processed by the other components of the network.
- Variable selection networks (14,7K, 14,7K and 1,8K parameters): these components are applied separately to the static, encoder and decoder variables. Their purpose is to allow the model to learn which predictors are more relevant at the different stages of the forecasting process.
- Gated Residual Networks (GRNs): these components are present throughout the architecture and provide a flexible mechanism for combining information from different sources while controlling the flow of information through gating mechanisms.
- LSTM encoder and decoder (2,2K parameters each): these components process the sequential information and capture temporal dependencies in the input sequence before the information is passed to the attention mechanism.
- Interpretable multi-head attention (1,1K parameters): this represents one of the main components of the TFT in terms of interpretability. Unlike standard multi-head attention, the interpretable version shares the attention weights across the different

heads, making it possible to examine which time steps receive greater importance in the forecasting process.

- Output layer (51 parameters): this final layer maps the resulting hidden representation into the forecast of the next-day EUR/USD return.

The model was trained using the Adam optimizer and the QuantileLoss function. Quantile loss is commonly used in TFT implementations because it allows the model to estimate multiple quantiles of the target distribution simultaneously. In the present analysis, however, the focus is on the median, or 50th percentile, forecast, which corresponds to the point forecast considered throughout the study. The quantile loss function allows over-predictions and under-predictions to be penalized differently depending on the selected quantile. For the median forecast, the loss is symmetric, making the resulting point prediction more comparable with the MSE-based evaluation used for the other models. The training procedure included early stopping based on the validation loss. The best model checkpoint was obtained at epoch 18, with a validation loss of approximately 0,1522. This result can be compared with the validation performance of the other deep learning models. The LSTM achieved a best validation loss of approximately 0,1575, but its test-set performance was considerably weaker, with an R^2 of -0,362. The TFT therefore achieved a slightly lower validation loss and, as discussed in the following section, considerably better test-set performance. This suggests that, in this case, the validation performance of the TFT provided a more useful indication of its out-of-sample generalization than was observed for the LSTM. The training process was relatively stable and the model converged within 18 epochs. This is comparable with the LSTM, which required 16 epochs, while it is considerably shorter than the MLP, which was trained for 54 epochs. The early stopping procedure therefore stopped the training relatively early, after the validation performance no longer showed further improvement. This behaviour may also be related to the regularization mechanisms incorporated into the TFT, including the variable selection networks and gated residual connections, which can help control the complexity of the model and reduce the risk of overfitting. Another technical aspect concerns the hardware used for training. The model was trained on the CPU rather than on a GPU, as indicated by the output showing “GPU available: False”. Although GPU acceleration would generally be preferable for a computationally

intensive architecture such as the TFT, the relatively small number of model parameters and the limited number of training epochs made CPU-based training feasible in this application. The training time remained manageable, and the absence of GPU acceleration did not prevent the completion of the hyperparameter tuning and the subsequent final evaluation. Regarding validation performance, the best validation loss of 0,1522 is slightly lower than the corresponding value obtained by the LSTM (0,1575). It is also important to compare this result with the MLP, whose validation loss reached approximately 0,1256, despite the clear signs of overfitting observed during training. Therefore, validation loss alone does not necessarily provide a complete indication of the model's final predictive performance. The independent test set remains the most appropriate basis for evaluating out-of-sample performance. The test-set results of the TFT are therefore discussed in the following section.

5.3.3 Variable Classification

A key feature of the Temporal Fusion Transformer is its ability to distinguish between static variables, time-varying known variables, and time-varying unknown variables. This distinction is particularly relevant in the present forecasting application because the model combines historical financial information with temporal variables whose future values are known in advance. Static variables are variables that remain constant over time for a given time series. In the present implementation, the explicitly defined static variable is “group_id”, which identifies the EUR/USD series. Since the dataset contains only one time series, all observations belong to the same group. Additional target-related features, such as the target scale and encoder length, are generated automatically by the “TimeSeriesDataSet” through “add_target_scales” and “add_encoder_length” and should therefore be interpreted as model-generated contextual features rather than as manually specified economic predictors. Time-varying known variables are variables whose values are available both during the historical observation period and over the forecasting horizon. In this analysis, they include “time_idx”, “day of week” (dow), “day of month” (dom), “week of year” (woy), and “month”. The relative time index is also generated automatically by the model through the “add_relative_time_idx” option. These variables can be used by the decoder because their values for the forecast horizon are known in advance. Time-varying unknown variables are variables whose future values are not available at the time the forecast is generated. In the present implementation, these

variables correspond to the financial variables included in “features_tft”, including the EUR/USD target and the other financial-market variables used as predictors. Their historical observations are available to the model, but their future values are not assumed to be known. They are therefore used primarily to provide historical information to the forecasting architecture. This classification is directly connected to the encoder-decoder structure of the TFT. The encoder processes the historical sequence available up to the forecast origin, using the past observations of the time-varying variables and the relevant temporal information. It therefore summarizes the information contained in the observed history. The decoder then operates over the forecasting horizon and incorporates variables whose future values are known in advance, particularly the calendar and temporal features. In the present application, the forecast horizon is one day, so the decoder can use the calendar characteristics of the next observation while it cannot use the future value of EUR/USD or the other financial variables, since these are unknown at the time of prediction. This separation allows the TFT to combine historical market information with legitimately available future information while avoiding the use of future financial observations that would not be available in a real forecasting setting.

5.3.4 Performance Plots

Figure 17 shows the TFT predictions compared with the observed EUR/USD returns on the test set. Compared with the LSTM, whose predictions were almost flat, the TFT is able to follow the general movements of the observed series more closely. The forecasts capture the central tendency of the actual returns and reproduce several of the medium-sized fluctuations. However, the model still has difficulty reproducing the largest positive and negative spikes. During periods of higher volatility, the predictions tend to smooth the most extreme movements, resulting in a series that is considerably less variable than the observed one. This behaviour is not specific to the TFT, but is also present in the other models considered in the analysis, suggesting that accurately forecasting extreme daily exchange rate movements represents a general challenge in this application.

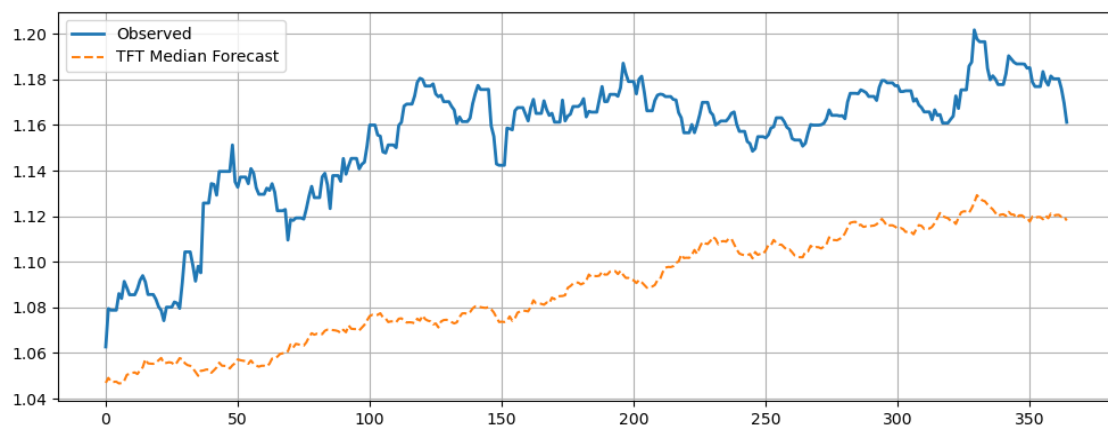


FIGURE 17 – TFT Predictions

An additional feature of the TFT is its ability to produce prediction intervals alongside point forecasts. Since the model was trained using quantile loss, it can estimate several quantiles of the conditional distribution of the target variable simultaneously. Figure 18 reports the 10th and 90th percentile prediction intervals (q10–q90), which define an interval expected to contain approximately 80% of the observations under the nominal coverage level. The empirical coverage of these intervals is 85,75%, which is slightly above the nominal 80%. This suggests that the prediction intervals are somewhat conservative, as they include a larger proportion of the actual observations than expected. The average interval width is 0,807, which is relatively large compared with the typical magnitude of the returns. This result reflects the high uncertainty associated with daily exchange rate movements and indicates that the model provides relatively wide intervals in order to account for this uncertainty. The directional accuracy of the TFT is 43,84%. This measure represents the proportion of observations for which the model correctly predicts the sign of the next-day return, that is, whether the EUR/USD return will be positive or negative. A prediction based on a random choice between the two directions would have an expected accuracy of approximately 50%, meaning that the value obtained by the TFT is below this benchmark. However, the result should be interpreted in the context of the generally limited predictability of daily financial returns. The fact that the directional accuracy remains relatively close to 50%, although slightly below it, is consistent with the results obtained from the other evaluation metrics. Similar values could also be expected for the linear and other machine learning models.

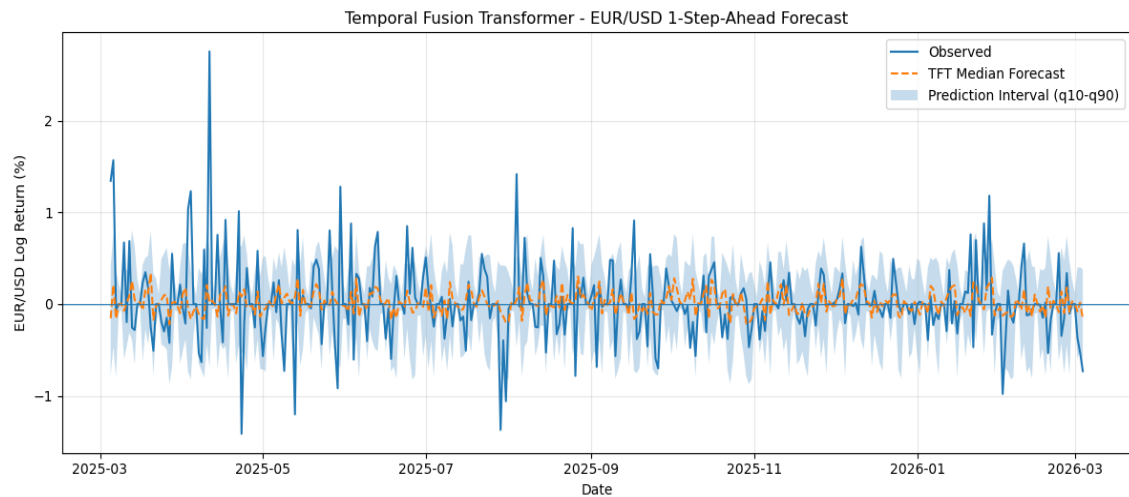


FIGURE 18 – TFT Prediction Interval

Another relevant aspect of the TFT results is the learning curve presented in Figure 19. The training loss decreases steadily during the first 15–18 epochs and subsequently reaches a relatively stable level. The validation loss follows a similar pattern, decreasing alongside the training loss without showing a substantial divergence between the two curves. This behaviour differs from that observed for the MLP, where the validation loss began to increase after approximately 20 epochs, and from the LSTM, which stopped after 16 epochs as a result of early stopping. The relatively similar behaviour of the training and validation losses suggests that the TFT was able to learn patterns from the training data without showing strong evidence of overfitting. The variable selection networks, gated residual connections and attention mechanism may have contributed to controlling the effective complexity of the model and maintaining a more stable training process.

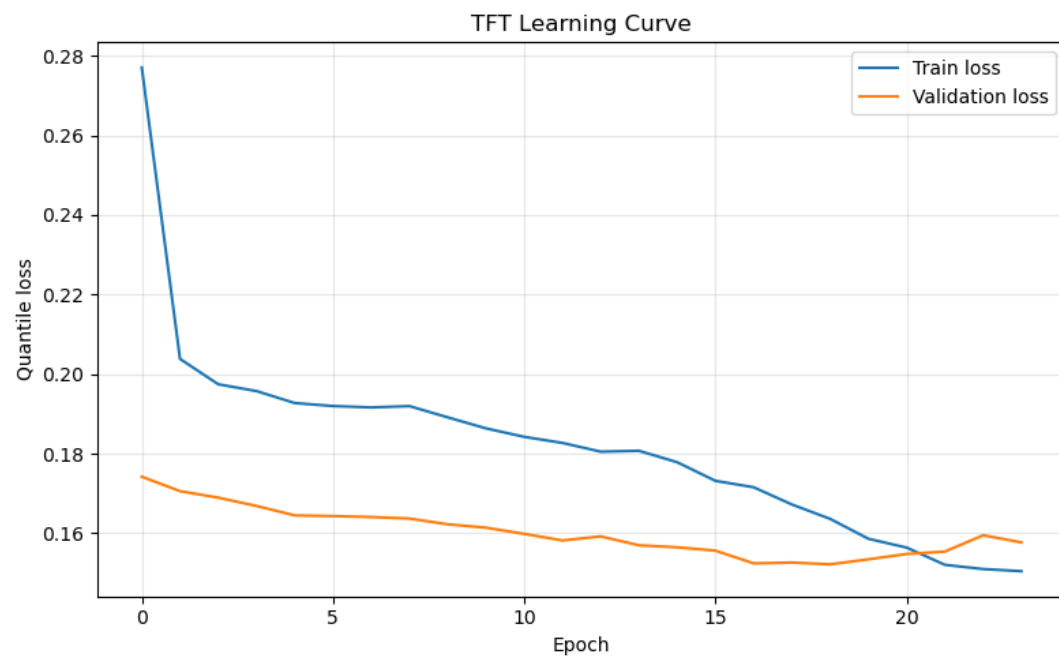


FIGURE 19 – TFT Learning Curve

5.3.6 Interpretation and Feature Importance

Figure 20 shows the attention weights over the time index, averaged across the test set. The attention mechanism in the TFT assigns a weight to each past time step, indicating how much importance the model gives to that observation when making a prediction. Higher weights therefore indicate that the corresponding time step receives more attention from the model. The pattern is quite clear. The attention weights are highest for the most recent time steps, that is, those closest to the prediction point, and generally decrease as we move further into the past. The weights remain relatively stable between approximately -60 and -20 time steps, where they fluctuate around 0,014 and 0,016. They then start to increase around -15 time steps, with a more pronounced increase after -10, eventually reaching their highest values, around 0,023 and 0,024, at the most recent time step (-1). This result is consistent with the idea that recent observations contain more information for predicting the next period. The ACF analysis presented in Section 1.4 showed that the linear autocorrelation in EUR/USD returns is relatively weak, with only a limited number of early lags showing significant autocorrelation. The attention weights provide a different but complementary perspective. Although the TFT does not assign an extremely large weight to any single observation, it gives greater importance to the most recent observations. This suggests that the model relies more strongly on recent information when generating its forecasts. At the same time, the attention weights do not fall to zero for the more distant observations. The model continues to assign non-negligible weights to observations going back approximately 50 or 60 days. Therefore, the TFT is not relying exclusively on the immediate past, but is also using information from a broader historical window, although with lower weights. This may indicate that information from earlier periods still contributes to the forecasting process, even if the most recent observations are given greater importance. Overall, the attention pattern suggests that the model combines information from both short- and longer-term periods. Recent observations receive the greatest weight, while observations further in the past continue to contribute to the prediction. This is one of the main advantages of the attention mechanism, since it allows the model to assign different levels of importance to different points in the input sequence rather than treating all time steps in the same way.

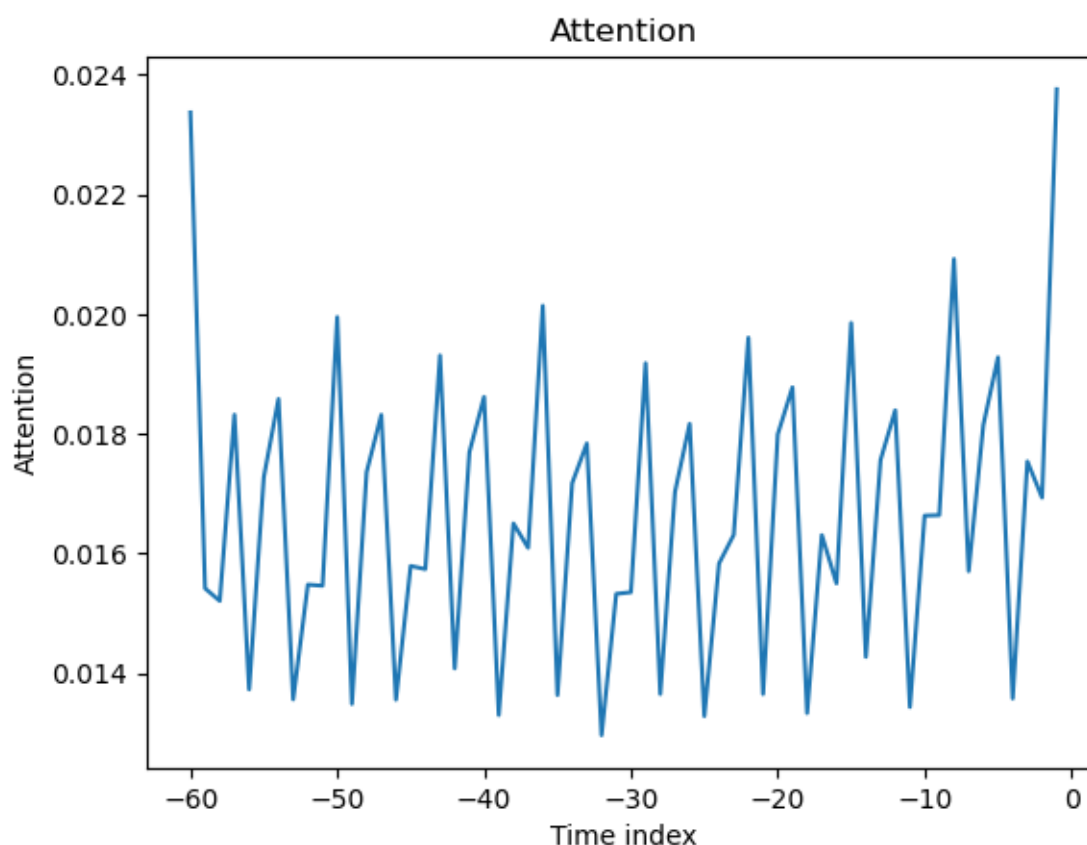


FIGURE 20 – Attention Weights

Figure 21 shows the importance assigned by the TFT to the variables included in the static component of the model. In the specific implementation adopted in this analysis, these variables include “EURUSD_raw_center”, “encoder_length”, “EURUSD_raw_scale”, and “group_id”. Although some of these variables are transformations or technical features rather than economically meaningful static characteristics, their importance scores provide information about how the model uses the different inputs when constructing the forecasting representation. The results reveal a clear concentration of importance among a small number of variables. “EURUSD_raw_center” accounts for approximately 55-60% of the total importance, followed by encoder_length, which contributes roughly 20-25%. “EURUSD_raw_scale” accounts for approximately 15-20%, while “group_id” has essentially zero importance. The relatively high importance assigned to “EURUSD_raw_center” is particularly noteworthy. This variable is directly related to the historical behaviour of the EUR/USD target and therefore provides the model with information about the recent dynamics of the dependent variable. Its dominant role is consistent with the attention results presented previously, where the TFT placed

substantial emphasis on recent observations. The model therefore appears to rely strongly on information contained in the recent history of the EUR/USD series when constructing its forecasts. The importance assigned to “EURUSD_raw_scale” is also relevant. This variable represents a standardized version of the same underlying EUR/USD information and therefore provides a different numerical representation of the target series. The fact that both the centred and scaled versions receive non-negligible importance suggests that the TFT may be exploiting different representations of the same underlying information when constructing its internal features. In particular, the centred variable reflects deviations from the mean, while the scaled representation expresses these deviations relative to the scale of the series. Moreover, “encoder_length” accounts for approximately 20-25% of the importance. Since the encoder length is determined by the forecasting configuration, its role should be interpreted primarily as a modelling or temporal-context feature rather than as an economically meaningful predictor. Its importance nevertheless indicates that the amount of historical information supplied to the model contributes to the representation used for forecasting. Finally, “group_id” receives essentially zero importance. This result is unsurprising because all observations belong to the same EUR/USD time series and therefore the group identifier does not provide meaningful variation across observations. Its negligible importance indicates that the model does not rely on this feature when generating its forecasts. Taken together, the static-variable results indicate that the TFT places substantial weight on information related to the historical behaviour and representation of the target variable. This finding is broadly consistent with the attention analysis and suggests that the recent history of EUR/USD returns provides an important component of the information used by the model.

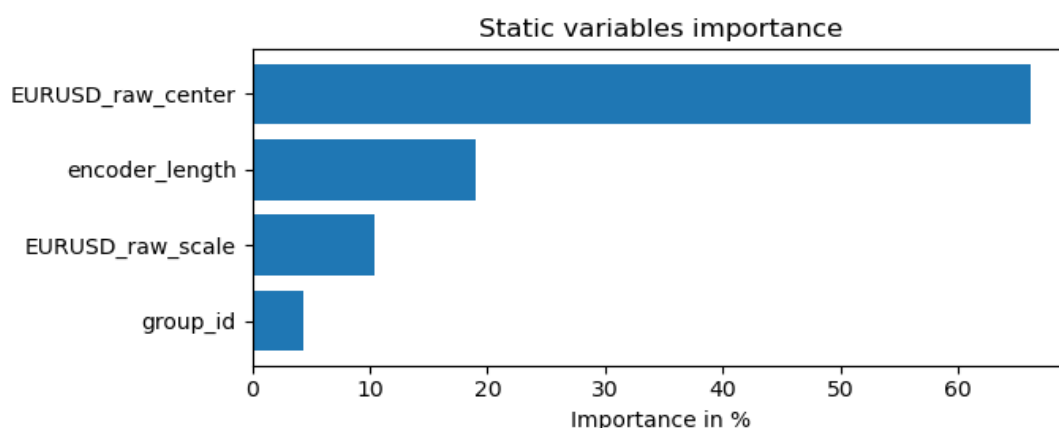


FIGURE 21 – Static Variables Importance

Figure 22 reports the variable importance scores for the encoder component of the TFT. The encoder processes the historical sequence of dynamic covariates, including the financial variables, the lagged target variable, and the calendar features available throughout the input sequence. The importance scores provide a ranking of the variables according to their contribution to the model's predictive representation. The results reveal a relatively clear hierarchy. The most important variable is the day of week (dow), which accounts for approximately 17,5% of the total importance. This is followed by Ethereum Price, with approximately 13,0%, the time index with approximately 11,5%, and Eurostoxx 50 Price with approximately 10,5%. These four variables alone account for approximately 52,5% of the total encoder importance. A second group of variables has importance values between approximately 5% and 7%. This group includes Gold Price at around 6,5%, Nasdaq Price at approximately 6,0%, the lagged EUR/USD variable at approximately 5,8%, Theta Price at around 5,5%, and Month at approximately 5,2%. Together with the variables in the first group, these features account for roughly 80% of the total importance. The remaining predictors therefore contribute considerably less to the encoder representation. One of the most notable findings is the relatively high importance assigned to calendar variables. The day of week is the single most important encoder variable, while the time index and month also receive substantial importance. Other calendar-related variables, including day of month, week of year, and relative time index, contribute additional information. Taken together, these results suggest that the temporal position of an observation provides useful predictive information to the TFT within the sample considered. This finding may be consistent with the existence of calendar-related patterns in financial markets. Different days of the week may be associated with different market conditions, liquidity patterns, positioning behaviour, or the timing of economic and financial information releases. However, the variable-importance results should not be interpreted as evidence of a causal day-of-week effect. Rather, they indicate that the model finds calendar information useful when constructing its forecasts. Ethereum is the most important financial variable in the encoder, with an importance of approximately 13,0%. While the relatively high importance of Ethereum suggests that the information contained in the cryptocurrency market contributes to the TFT's forecasts of EUR/USD returns within the considered sample. This does not imply a causal relationship between Ethereum and EUR/USD, but rather indicates that the

model finds Ethereum useful for improving its predictions. The equity indices also remain important predictors. Eurostoxx 50 Price accounts for approximately 10,5% of encoder importance, while Nasdaq Price contributes approximately 6,0%. Gold Price accounts for around 6,5. The lagged EUR/USD variable receives an importance of approximately 5,8%. This is noteworthy given the weak linear autocorrelation identified in the preliminary analysis. The result suggests that, although the target series may contain limited linear dependence on its past values, the TFT is nevertheless able to extract some predictive information from the historical EUR/USD sequence. This is consistent with the idea that nonlinear predictive relationships may exist even when linear autocorrelation is relatively weak.

The cryptocurrency variables also display a clear hierarchy. Ethereum and Theta receive relatively high importance scores, while Bitcoin contributes approximately 4,5%. By contrast, the stablecoin variables, including Dai, USD Coin and Tether, receive substantially lower importance scores. Their low importance indicates that, within the information set and sample considered, these variables provide relatively little additional predictive information to the TFT compared with the more important financial and calendar features. The traditional exchange-rate variables occupy relatively low positions in the ranking. GBP/USD, CNY/USD, CHF/USD, NZD/USD, BRL/USD and RUB/USD all receive considerably lower importance than the main financial predictors. Overall, the encoder importance results suggest that the TFT relies on a combination of temporal information and financial-market variables. Calendar features, particularly the day of week, play a substantial role, while Ethereum, Eurostoxx 50, Gold and Nasdaq represent the most relevant financial predictors. At the same time, several stablecoins and exchange-rate variables receive relatively little importance.

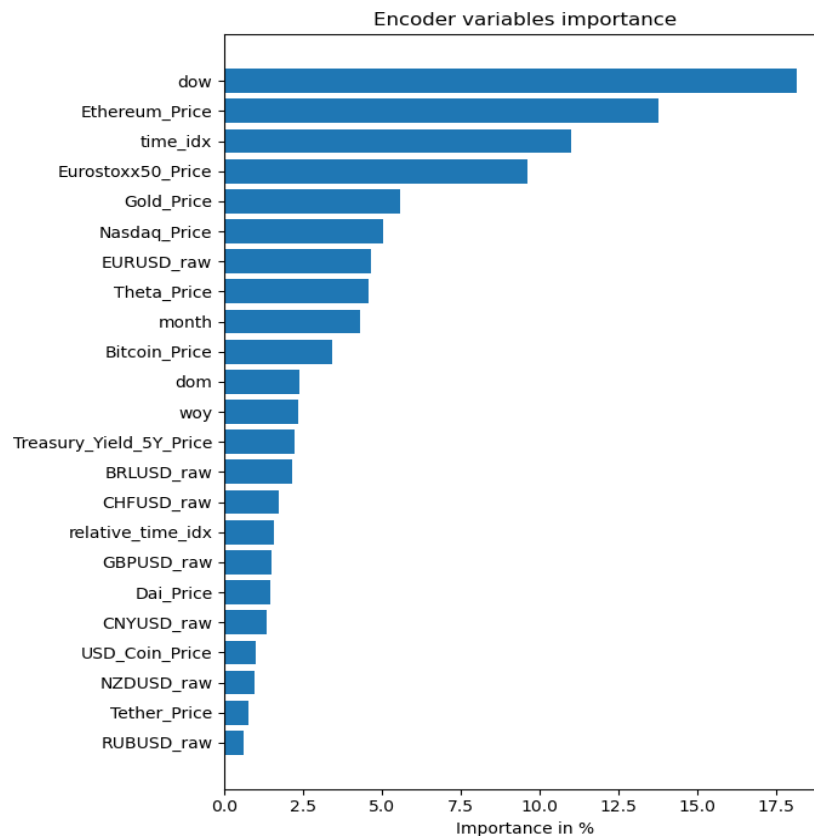


FIGURE 22 – Encoder Variables Importance

Finally, figure 23 reports the variable importance scores for the decoder component of the TFT. In the present one-step-ahead forecasting setting, the decoder primarily processes information associated with the forecasting horizon and future-known temporal variables, such as the day of week, month, week of year, day of month and relative time index. The results show a particularly strong concentration of importance in the calendar variables. The day of week (dow) dominates the decoder importance, accounting for approximately 45-50% of the total. This is followed by “relative_time_idx”, which contributes approximately 25-30%. Month and week of year each account for approximately 5-10%, while day of month contributes around 5%. The time index has the lowest importance, at approximately 2-3%. The dominance of the day-of-week variable is particularly noteworthy because the same variable was also the most important feature in the encoder, although with a considerably lower importance of approximately 17,5%. Its much larger contribution in the decoder suggests that calendar information plays an especially important role when the model constructs the final forecast for the next time step. This result suggests that the TFT has identified systematic differences associated

with the timing of the forecast within the trading calendar. For example, the model may assign different expected patterns to forecasts made for different weekdays. Such patterns could potentially reflect differences in liquidity, market positioning, information flows or the timing of economic announcements. However, as with the encoder results, the importance assigned to the day of week should be interpreted as evidence of predictive usefulness within the model rather than as proof of a specific economic or causal effect. The relatively high importance of “relative_time_idx”, which accounts for approximately 25-30% of the decoder importance, also deserves attention. This variable represents the relative position of the observation within the forecasting horizon. Since the present analysis focuses on one-step-ahead prediction, its economic interpretation is necessarily limited. Its relatively high importance should therefore be understood primarily as an indication that the model uses information about the temporal position of the forecast when constructing its decoder representation. Month and week of year receive more moderate importance, each accounting for approximately 5-10%. Their presence suggests that the TFT is able to incorporate calendar information operating at different temporal frequencies. While the day of week captures short-term calendar variation, month and week of year may allow the model to account for broader seasonal or recurring temporal patterns. The day-of-month variable contributes approximately 5% to the decoder representation. Although its importance is considerably lower than that of the day of week, it remains non-negligible and suggests that the position within the monthly calendar may provide some additional predictive information. The time index has the lowest importance in the decoder, at approximately 2-3%. This indicates that the absolute position in the sample contributes relatively little to the final decoder representation compared with the more specific calendar features. The model therefore appears to rely more heavily on recurring temporal characteristics than on the absolute passage of time.

Comparing the decoder and encoder results reveals an interesting distinction. In the encoder, financial variables such as Ethereum, Eurostoxx 50, Gold and Nasdaq are among the most important predictors, together with calendar variables. In the decoder, by contrast, the importance is concentrated almost entirely among temporal and calendar features. This difference reflects the different roles of the two components: the encoder processes the historical information available to the model, while the decoder incorporates the information associated with the forecasting horizon and future-known

variables. Overall, the decoder importance results reinforce the central role of calendar information in the TFT. In particular, the strong importance assigned to the day of week suggests that the temporal context of the forecast is an important component of the model's prediction process. This is an interesting finding because the earlier linear and machine-learning specifications did not explicitly provide the same type of temporal representation.

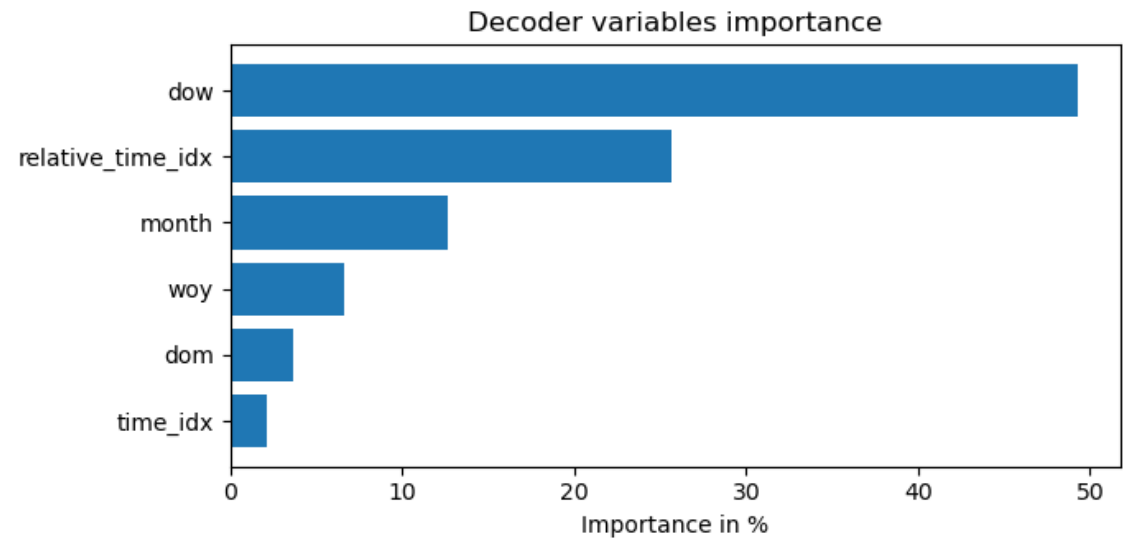


FIGURE 23 – Decoder Variables Importance

6 CONCLUSIONS

6.1 Summary of Findings

This thesis investigated the predictability of daily EUR/USD returns by comparing a broad range of econometric, nonlinear, machine learning, and deep learning models within a common out-of-sample forecasting framework. The main objective was to assess whether increasing the flexibility and complexity of the forecasting models could lead to more accurate predictions of the next-day EUR/USD return, while also examining which financial variables and temporal characteristics contain useful predictive information. The empirical analysis followed a clear progression: from simple linear models (OLS, Ridge, Lasso), to nonlinear parametric specifications (Polynomial and Spline regression), to tree-based and kernel-based machine learning algorithms (Random Forest, XGBoost, SVR), and finally to deep learning architectures (MLP, LSTM, TFT). Each stage provided useful insights into the forecasting problem, although the overall results lead to a nuanced conclusion. The linear models provided a relatively modest baseline, with out-of-sample R^2 values ranging between approximately 2% and 3%. The nonlinear parametric models, particularly Spline regression, produced a noticeable improvement, achieving an R^2 of approximately 9%. This result suggests that allowing for nonlinear relationships between the predictors and the EUR/USD return can provide useful additional information compared with a purely linear specification. The machine learning models, especially XGBoost ($R^2 = 7,2\%$), further supported the importance of nonlinearities and interactions, although their performance remained below that of the spline specification. The results obtained from the deep learning models were more mixed. The MLP failed to achieve a positive R^2 ($-0,014$), while the LSTM performed considerably worse ($-0,362$), producing predictions that were largely concentrated around the mean of the target series. The TFT, however, represented a clear improvement within the deep learning group, achieving an R^2 of 4,4%, the lowest RMSE among the deep learning models (0,399), and the lowest MAE overall (0,245). In addition to its predictive performance, the TFT provided useful interpretability through its attention mechanisms and variable selection networks, offering insights into the variables and temporal characteristics that contributed most to its forecasts. Perhaps the most relevant finding of

the analysis is not the identification of a single superior model, but rather the consistent pattern observed across the different specifications: the out-of-sample predictability of daily EUR/USD returns remains limited, regardless of the modelling approach adopted. The best-performing model in terms of overall explanatory power was Spline regression ($R^2 = 9,0\%$), while XGBoost also achieved relatively strong performance ($R^2 = 7,2\%$). Both models are considerably simpler than the deep learning architectures considered in the final stage of the analysis. This suggests that, for the forecasting problem and dataset considered in this study, increasing model complexity does not necessarily lead to better predictive performance. To provide a clear overview of the results, the following table summarises the out-of-sample performance of all models considered in this analysis.

Model	MSE	RMSE	MAE	R^2
OLS	0,1629	0,4036	0,2607	0,0218
Ridge	0,1614	0,4018	0,2586	0,0308
Lasso	0,1612	0,4015	0,2574	0,0323
Polynomial	0,2644	0,5142	0,3009	−0,5875
Spline	0,1515	0,3892	0,2532	0,0905
Random Forest	0,1568	0,3960	0,2506	0,0585
XGBoost	0,1546	0,3931	0,2503	0,0721
SVR (RBF)	0,1608	0,4010	0,2514	0,0347
MLP	0,1690	0,4111	0,2628	−0,0144
LSTM	0,2130	0,4615	0,2607	−0,3624
TFT	0,1592	0,3990	0,2452	0,0442

Several patterns emerge from this comparison. Firstly, the linear models perform similarly, with regularization providing only modest improvements. Moreover, the nonlinear parametric models show a clear distinction: Polynomial Regression fails, while Spline Regression achieves the highest R^2 among all models (9,0%) and the lowest RMSE (0,389). Furthermore, the machine learning models offer consistent improvements over the linear benchmarks, with XGBoost emerging as the strongest performer in this category. The best-performing model depends on the evaluation metric. If minimizing

RMSE is the primary objective, Spline Regression is the best choice ($RMSE = 0,389$). If minimizing MAE is more important, the TFT is preferable ($MAE = 0,245$). If maximizing R^2 is the goal, Spline Regression again performs best ($0,090$). XGBoost offers a good balance across all metrics. The LSTM is the clear worst performer, with an R^2 of $-0,362$ indicating that its predictions are substantially worse than a simple mean forecast.

6.2 Implications for Exchange Rate Forecasting

The results of this analysis have several implications for the forecasting of exchange rate movements. First, the limited predictive power obtained across all models confirms the well-known difficulty of forecasting daily exchange rate returns. The weak linear autocorrelation identified in the preliminary analysis, together with the consistently low R^2 values, suggests that daily EUR/USD returns contain only a limited amount of predictable information, at least with respect to the set of predictors considered in this study. Second, the results indicate that greater model complexity does not necessarily lead to better forecasting performance. The LSTM, despite being specifically designed to capture sequential dependencies, performed considerably worse than several simpler models. This finding highlights an important limitation of deep learning approaches in financial forecasting. Their effectiveness depends not only on the flexibility of the architecture, but also on the availability of sufficient data, an appropriate model structure, and adequate hyperparameter tuning. When these conditions are not fully satisfied, simpler models may provide more reliable out-of-sample forecasts than considerably more complex alternatives. Third, the results obtained with the TFT highlight the potential importance of interpretability in financial forecasting. Although its R^2 remained below that of Spline regression and XGBoost, the TFT provided additional information through its attention mechanisms and variable importance measures. In particular, the importance assigned to calendar variables, such as the day of the week, suggests that temporal context may contain relevant information for EUR/USD forecasting, a feature that was not as evident in the traditional regression models. In addition, the prediction intervals generated by the TFT provide a useful measure of forecast uncertainty that cannot be obtained from point forecasts alone. Fourth, the importance attributed to calendar variables, particularly within the TFT, suggests that exchange rate movements may exhibit systematic patterns associated with the trading calendar. Although these

results should be interpreted with caution, they indicate that future forecasting models could benefit from explicitly incorporating calendar-based variables. Such features may provide additional information about market behaviour that is not captured by financial and macroeconomic predictors alone. Fifth, the relatively high importance assigned to cryptocurrency variables, particularly Ethereum in the TFT, suggests that digital assets may contain predictive information for traditional foreign exchange markets. This result is particularly interesting given the increasing interaction between cryptocurrency and traditional financial markets. Although the present analysis does not establish a causal relationship, the findings suggest that the role of digital assets in exchange rate forecasting deserves further investigation. Finally, from a practical perspective, the results suggest that model selection and feature engineering may be more important than simply increasing architectural complexity. Spline Regression and XGBoost, both relatively straightforward compared with the deep learning approaches, achieved performance that was comparable to or better than the more sophisticated neural network models. This represents an important consideration for practitioners, particularly when computational resources, interpretability, and implementation complexity are relevant constraints.

6.3 Economic Implications

The empirical findings of this study carry several implications for both the academic understanding of exchange rate dynamics and the practical conduct of forecasting in financial markets. Firstly, the consistently limited predictive power observed across all models reinforces the view that daily EUR/USD returns are predominantly driven by unpredictable news and market sentiment rather than by systematic relationships captured by observable financial variables. This is consistent with the Efficient Market Hypothesis in its weak form, which posits that asset prices already incorporate all available historical information, making short-term price movements inherently difficult to forecast. However, the fact that some models (particularly Spline Regression and XGBoost) achieved positive predictive performance suggests that market efficiency is not absolute and that certain nonlinear relationships, interactions, and calendar effects can be exploited to obtain a marginal forecasting advantage. The results suggest that the incremental value of sophisticated forecasting models for daily exchange rate trading strategies is likely limited. The modest R^2 improvements, combined with the negative performance of

several deep learning architectures, imply that the economic gains from using complex models may not justify their implementation costs; unless accompanied by appropriate risk management frameworks or used in conjunction with other sources of information. In this sense, simpler models such as Spline Regression and XGBoost, which offer comparable or superior performance to the TFT while being more transparent and computationally efficient, may represent a more prudent choice for market participants. Another important implication concerns the role of cryptocurrency markets, particularly Ethereum, which emerged as a relevant predictor in the TFT and in the tree-based models. This finding suggests that the growing integration between digital assets and traditional financial markets may have spillover effects that are not yet fully understood. It raises the question of whether cryptocurrency price movements reflect shifts in global risk appetite, liquidity conditions, or speculative flows that subsequently affect major currency pairs (an area that warrants further investigation, particularly as digital assets continue to gain institutional acceptance). The prominence of calendar features across multiple models also has economic relevance. If systematic differences exist between, say, Monday and Friday returns, this could reflect institutional trading patterns, such as end-of-week position squaring, the timing of economic data releases, or differences in market liquidity. These patterns, if persistent, may offer exploitable opportunities for short-term trading strategies, though their economic magnitude appears limited given the low R^2 values. The ability of the TFT to provide prediction intervals alongside point forecasts represents a further practical advantage, as it allows decision-makers to quantify the uncertainty surrounding their forecasts. In risk management, for instance, understanding the range within which the next-day return is likely to fall is often more valuable than a single point estimate. The TFT's interval coverage of approximately 86% (against a nominal 80%) suggests that its uncertainty estimates are reasonably well-calibrated, which could be useful for Value-at-Risk calculations or for setting stop-loss levels in trading operations. Finally, the overall conclusion that model complexity does not guarantee better predictive performance has a broader economic implication: it highlights the importance of distinguishing between in-sample fit and out-of-sample generalisation. The poor performance of the LSTM, in particular, serves as a cautionary example of how sophisticated architectures can fail when the underlying signal is weak and the data are limited. In such contexts, the principle of parsimony (favouring simpler models that

capture the essential structure of the data without overfitting) remains a sound guiding principle for both academic research and practical forecasting.

6.4 Limitations and Future Research

Several limitations should be considered when interpreting the results of this study. Firstly, the analysis covers approximately five years of daily observations, from March 2021 to March 2026. Although this provides a substantial number of observations for traditional econometric models, the effective sample size remains relatively limited for more complex deep learning architectures. The LSTM results in particular suggest that model complexity may exceed the amount of stable information available in the sample. Moreover, the explanatory variables considered in the analysis mainly consist of financial market variables. Although these variables provide a broad representation of market conditions, EUR/USD movements are also affected by macroeconomic announcements, monetary policy decisions, inflation, employment conditions, interest-rate expectations, geopolitical events and changes in investor sentiment. Incorporating such information could potentially improve the forecasting performance of the models by providing variables that more directly capture the fundamental drivers of exchange-rate movements. Additionally, the analysis focuses on one-step-ahead forecasting of daily returns. This horizon is particularly challenging because financial markets incorporate new information rapidly and daily returns contain relatively weak linear dependence. Future research could therefore investigate alternative forecasting horizons, such as intraday, weekly or multi-day forecasts, and examine whether the relative performance of the models changes when the forecasting horizon is modified. A further extension could involve the use of rolling or expanding-window estimation procedures. The current chronological train-validation-test framework provides a clear and rigorous evaluation, but financial markets are characterized by structural changes and evolving relationships. A rolling evaluation could therefore provide additional information about the stability of model performance across different market regimes. It would also be interesting to investigate whether models perform differently during periods of high and low volatility. Finally, future research could explore ensemble approaches that combine the predictions of models with complementary strengths. The results of this thesis suggest that different models capture different aspects of the forecasting problem: linear regularized models

provide interpretability and variable selection, spline and tree-based methods capture nonlinear relationships, while the TFT provides a richer representation of temporal dependencies and predictive uncertainty. Combining these approaches could potentially produce forecasts that are more robust than those generated by any single model.