[1]

Machine learning in foreign exchange

Bo Yuan[a]

[a] *Judge Business School, University of Cambridge, Cambridge*

Abstract

We apply machine learning to predict currency excess return cross-sectionally while addressing the black-box issue through interpretability techniques. First, neural networks (NN) significantly outperform traditional models in predicting currency excess returns including the random walk, highlighting the advantages of more flexible predictive functions. Second, NN-based portfolios achieve higher Sharpe ratios, underscoring their economic value. Third, both local and global interpretability techniques reveal that interactions between global macroeconomic factors and currency-specific characteristics are key drivers of FX risk premia.

Keywords: Foreign Exchange Markets, Cross Section of Returns, Machine Learning, Interpretability Analysis

[1] We thank Lucio Sarno, Andrei Kirilenko, Murillo Campello, Mohammad Hashem Pesaran, Bart Lambrecht and audience participants at International Symposium on Forecasting and Cambridge University. The opinions here expressed are solely those of the authors and do not represent in any way those of their employers.

Email address: by258@cam.ac.uk (Bo Yuan)

1. Introduction

This paper applies machine learning techniques to analyze the cross-section of currency excess returns. We investigate the following questions: (1) Can machine learning methods enhance the predictability of excess returns beyond traditional linear models? (2) Do machine learning models provide economically meaningful improvements in understanding currency excess returns? (3) Does interpretability analysis of machine learning reveal which factors—tradable or nontradable—are most relevant, and how their interactions shape return variation?

Our research presents machine learning and other conventional econometric models to explain currency excess returns in the cross-section. Our contributions primarily lie in the following three aspects. First, we employ more advanced models to address the limitations of conventional linear factor pricing models. Empirical asset pricing has undergone significant evolution, from the foundational Capital Asset Pricing Model (CAPM) (Sharpe, 1964; Lintner, 1965) and Arbitrage Pricing Theory (APT) (Ross, 1976) to the subsequent proliferation of risk factors. Despite these advancements, empirical challenges remain. We find ourselves amidst the era of high-dimensional factors. The high-dimensional nature of potential predictors introduces significant model uncertainty, making it difficult to determine the true drivers of excess returns. Traditional econometric methods often struggle with factor selection, dimension reduction, nonlinear interactions, and robustness in out-of-sample forecasts (Gu et al., 2020). The existing literature provides relatively little guidance on predictor and prediction function (Giglio et al., 2022). It is unclear whether the functional form is linear and which are the true predictors. Machine learning is a good tool to solve this problem. On the one hand, it can handle very large predictor sets very well. At the same time, there are many regularization methods to deal with overfitting. On the other hand, it allows more and more flexible functional forms.

Second, we provide further insights into the factor zoo problem in currency markets. For the currency market, understanding the trade-off between the risk and return is crucial for both academic research and practical investment strategies. Following the work of Lustig and Verdelhan (2007), studies have increasingly examined the cross-section of currency excess returns through the lens of risk factors. This has led to a growing literature on tradable factors, derived from currency investment strategies such as carry (Lustig et al., 2011) and momentum (Menkhoff et al., 2012b), and nontradable macro-financial factors (Nucera et al., 2024). However, this expansion of risk

factors—the so-called "factor zoo"—raises fundamental questions about which sources of risk best explain currency excess returns and how their interactions shape return dynamics.

Many existing studies focus primarily on exchange rate movements rather than directly modeling currency excess returns. The latter is more relevant to risk-based factor pricing and systematic investment strategies. In this regard, machine learning can provide a data-driven approach to learning about predictive returns. For example, it can tackle well large predictor sets and can capture nonlinear relationships (Gu et al., 2020). In addition, machine learning has been widely used in the stock market. Unlike prior studies that focus on either investment-strategy-based factors or macro-financial risk factors separately, we examine both jointly, providing a more comprehensive perspective on FX pricing. However, its potential in the foreign exchange market remains unexploited, possibly because economic interpretability of machine learning models remains an unsolved issue. This limits their application in empirical asset pricing.

Third, we introduce new interpretability tools to address the black-box problem in machine learning, offering a better understanding of the drivers of currency excess returns. We adapt common methodologies from the engineering domain to address the black-box issue in finance. On the one hand, we employ local interpretability techniques, such as DeepLIFT and Layer-wise Relevance Propagation (LRP), which analyze each sample instance individually before averaging across instances. On the other hand, for global interpretability, we leverage Shapley values to identify the most significant predictors, reveal intricate return dynamics, and enhance return forecasting.

Our findings reveal several key insights. First, neural networks outperform the linear models and the tree-based models. It is the only model that beats the random walk in predicting the cross section of currency excess returns. However, as the forecast horizon increases from one month to twelve months, their predictive power decreases, which is consistent with the result in the open economy (Hassan et al., 2024). Second, the interaction between global conditions and currency-specific characteristics plays a crucial role in shaping excess returns, emphasizing the need to jointly consider tradable and nontradable factors. Third, ML-based portfolios achieve superior risk-adjusted returns, with higher Sharpe ratios compared to conventional factor-based approaches. Finally, we introduce interpretability techniques such as Shapley value analysis to decompose the contributions of different predictors, offering novel insights into the economic drivers of currency risk premia. The results vary across prediction horizons, but a key insight is that the interaction between global

market conditions and currency-specific characteristics jointly shapes return dynamics.

A related strand of literature investigates the application of machine learning in equity markets (Gu et al., 2020). Building on Filippou et al. (2023), we expand the set of predictors beyond the previously considered 70 macroeconomic and country-specific variables by incorporating tradable factors and economic state indicators. Additionally, while prior studies focused on only developed markets, we extend our analysis to a broader set of 49 countries, thereby providing a more comprehensive assessment of global return predictability. Our study also contributes by introducing novel interpretability techniques to better understand ML-driven predictions and by implementing a wider range of predictive models beyond conventional linear regressions and deep neural networks. Moreover, we address key concerns regarding the implementability of ML-based trading strategies, a topic of ongoing debate in finance. Furthermore, we explore the role of risk drivers in foreign exchange (FX) returns and their connection to the broader factor zoo literature, drawing parallels with the three-pass regression approach (Nucera et al., 2024). Our findings contribute to the literature by demonstrating that ML (neural networks) outperforms traditional models in return forecasting, reinforcing its potential as a powerful tool in empirical asset pricing.

The structure of this paper is as follows. Section 2 reviews the relevant literature. Section 3 describes the data, including currency-specific characteristics and global factors. Section 4 outlines the methodology, detailing the model specification and evaluation framework. Section 5 presents the prediction results, Section 6 discusses the economic performance, and Section 7 further explores the interpretability of the model. Finally, Section 8 summarizes the key conclusions and discusses the broader implications of the findings.

2. Literature review

Since Gu et al. (2020) introduced machine learning into asset pricing, research has flourished in two directions. On the one hand, econometricians outside the market use machine learning in the context of Stochastic Discount Factor (SDF) extraction in high-dimensional settings. Kelly et al. (2019) proposed Instrumented Principal Component Analysis, enabling time-varying loadings on latent factors. On the other hand, investors within the market use it for prediction. Gu et al. (2020) applied various machine learning techniques to the US stock market, while Leippold et al. (2022) extended the analysis to the Chinese stock market. Chen et al. (2024) combined Genera-

tive Adversarial Network, Recurrent Long Short-Term Memory Network, and Feedforward Neural Network to explain cross-sectional return differences. Bianchi et al. (2021) examined predictable variation in bond returns with machine learning.

A key question in international finance is the predictability of exchange rates. Meese and Rogoff (1983) found that macroeconomic models fail to outperform a random walk over short horizons, known as the "Macro Exchange Rate Disconnect Puzzle." Fama (1984) documented the "Forward Premium Puzzle," where forward premiums do not predict future spot exchange rates but reflect a risk premium (Hansen and Hodrick, 1980). Lustig et al. (2011) found high-interest-rate currencies earn higher excess returns. Menkhoff et al. (2012b) revealed that currencies that have performed the best over the last three to twelve months typically continue to generate higher returns. Menkhoff et al. (2017) identified a value factor, where undervalued currencies yield higher future returns, though its predictive power emerges only after 6 to 24 months.

It is natural to examine its effectiveness in FX markets, where traditional models (e.g., Mark (1995); Engel et al. (2007)) test whether fundamentals predict future exchange rates. Recent studies explore more flexible, data-driven models. Wada (2022) shows that band spectral regression and LASSO outperform linear benchmarks out-of-sample. Amat et al. (2018) use machine learning to assess short-horizon FX predictability, finding that time-varying, nonlinear effects of macro fundamentals help improve forecasts. Filippou et al. (2020) implement sequential ridge regression and exponentially weighted averaging with discounting, showing these models can outperform the random walk when modeling short-term links between fundamentals and FX returns. Pfahler (2021) incorporate interaction terms between fundamentals and time dummies, finding that ANN and XGBoost significantly boost predictive accuracy relative to standard models. Yaohao and Albuquerque (2019) test 90 SVR models across 10 currencies using macro variables and different Kernel functions; while most outperform the Random Walk in point forecasts, only 40% show statistically significant improvements.

Most machine learning studies on currency focus on predicting exchange rates, not currency excess returns. Our research shifts this focus, investigating the drivers of currency excess returns—distinct from the well-known currency disconnect puzzle (Meese and Rogoff, 1983). While ML has shown promise in other financial markets, its effectiveness in FX remains uncertain. We aim to bridge this gap by systematically testing ML's ability to uncover insights into currency

excess returns. Unlike Filippou et al. (2020), which relies on macro fundamentals and addresses time-varying loadings but overlooks tradable style factors, we take a cross-sectional approach, emphasizing predictive drivers beyond the macro disconnect perspective.

However, one fundamental challenge of applying machine learning in financial markets is its black-box nature. Unlike traditional econometric models, ML techniques often lack clear economic interpretability, making it difficult to attribute predictive power to economic fundamentals. Some studies, such as Gu et al. (2020), propose methods to enhance interpretability. Given this concern, this study employs model-agnostic interpretability techniques, including Shapley values (Lundberg and Lee, 2017).

We contribute to the growing FX "factor zoo" literature by examining cross-sectional drivers of currency excess returns. While much work focuses on tradable factors like carry (Lustig et al., 2011) and momentum (Menkhoff et al., 2012b), recent studies highlight non-tradable risks (Nucera et al., 2024). This study jointly considers both types and their interactions, offering a more complete view of the FX risk–return trade-off.

We focus on conditional expected returns, which incorporate current information, unlike unconditional historical averages. Following Chernov et al. (2023), who advocates a conditional SDF approach, we integrate market conditions and currency-specific traits to reflect the timing nature of FX strategies. Results show that conditional information is essential for forecasting currency returns.

3. Data

Our input, the predictor set, for each currency pair includes a number of characteristics, and interactions of each characteristic with global time-series variables.

3.1. Individual characteristic

As per the data used for setting up the individual characteristic for each of 49 currency pairs, here is all the information, including the characteristic name, the construction approach, and the data sources.

Currency excess return

Define the spot rate as S_t and the corresponding one-month forward rate (midquote) as F_t for each currency pair. The data is from Reuters and WM/Reuters accessed via Datastream and Barclays

Bank International (BBI). The exchange rate is defined as the number of USD per unit of foreign currency. Unless otherwise stated, all returns are expressed in log terms. The currency excess return is derived from purchasing foreign currency in the forward market at time t and selling it in the spot market at time $t+1$. This return captures the difference between the forward rate set at t and the realized spot rate at $t+1$. This can be calculated as follows:

$$r_{t+1} = log(S_{t+1}) - log(F_t),$$

This is equivalent to the spot exchange rate return minus the forward premium:

$$r_{t+1} = (log(S_{t+1}) - log(S_t)) - (log(F_t) - log(S_t)).$$

The Covered Interest Parity (CIP) condition says the forward premium approximately equals the interest rate differential:

$$log(F_t) - log(S_t) \simeq i_t - i_t^*,$$

where i_t and i_t^* are the risk-free rates in the domestic and foreign country respectively, over the forward contract maturity. If CIP holds, the currency excess return is approximately equal to the spot exchange rate return plus the interest rate differential relative to the domestic country:

$$(log(S_{t+1}) - log(S_t)) + (i_t^* - i_t).$$

Carry

We use $log(S_{it}) - log(F_{it})$, where S and F are the correspondingly spot and one-month forward exchange rate quotes (Menkhoff et al., 2012a), i and t correspondingly refer to the currency pair and the month.

Short-term Momentum

We use the currency excess return over the previous month (Menkhoff et al., 2012b). The calculation is based on spot and forward exchange rate quotes (midquote).

Long-term Momentum

We use the currency excess return over the previous 12 months skipping the last month. Although long-term momentum in the currency market does not necessarily require skipping the most recent

month of returns, we adopt this approach to ensure consistency (Asness et al., 2013). Moreover, momentum returns for currencies are actually stronger when the most recent month is included, making our results more conservative.

Currency Value

For currencies, the value factor is defined as the change in the real exchange rate over the past five years, which should be calculated as $log(Q_t)\check{}log(Q_t - 5)$, where Q is defined as the real exchange rate as below (Menkhoff et al., 2017): $Q_t = \frac{P_t}{P_t^* S_t}$, where S denotes the exchange rate (USD per unit of foreign currency), P denotes the US price level, and P^* denotes the foreign price level. Consumer Price Index (CPI) data is from IMF International Financial Statistics, except for Taiwan from National Statistics.

Net Foreign Assets

We use $\frac{-NFA}{GDP}$ following (Della Corte et al., 2016).

Long-term Yields

We use $\left(i_{10yr} - i_{10yr}^{US}\right)$, say, the difference between the foreign country's 10-year interest rate and the corresponding rate in the the United States. The calculation uses the interest rates available on OECD Monthly Monetary and Financial Statistics.

Term Spread

We use $(i_{10yr} - i_{3mo})$, say, foreign country's term spread defined as the difference between the 10-year and 3-month rates. The Long (Short)-term interest rates comes from OECD Monthly Monetary and Financial Statistics.

3.2. Global characteristic

In our study, we construct the global predictor set by referencing the framework established in Nucera et al. (2024), which details the development of a comprehensive global non-tradable characteristic set. To enhance the model's robustness and capture nuanced relationships, We further expand the characteristic set by incorporating the interactions between seven distinct currency-level characteristics and a range of global state variables. Specifically, this involves introducing cross-terms where each currency-specific attribute is interacted with various aggregate time-series indicators that reflect broader market conditions. As a result, the predictor set is significantly enriched, encompassing a total of seven primary characteristics for each currency pair alongside their interaction terms with the global variables. Altogether, this methodology generates a characteristic

Table 1: **Summary Statistics**

This table presents the monthly prediction performance at the currency level, comparing in-sample and out-of-sample correlation values. The data is divided into training, validation, and testing samples. The model is trained and validated on the training and validation data, respectively, and then applied to the test data without further adjustment. The evaluation includes five models: Ordinary Least Squares (OLS), OLS with Huber loss (OLSH), neural networks (NN) with increasing complexity, from one to eight hidden layers, the dimension reduction linear models including Partial Least Squares (PLS) and Principal Component Regression (PCR), as well as the tree-based models including Random Forest (RF) and Gradient Boosted Regression Trees (GBRT). Correlation, as a bounded measure ranging from -1 to +1, quantifies the strength and direction of the linear relationship between predicted and actual values. A correlation of +1 signifies a perfect positive linear relationship, where the model's predictions move in exact proportion to the observed values. Conversely, a correlation of -1 indicates a perfect negative linear relationship, meaning the model systematically predicts the opposite of the actual outcomes. A correlation close to zero suggests little to no linear relationship, implying that the model's predictions have limited explanatory power for the observed data.

	Mean	Median	Std	Skew	Min	Max	Kurtosis
carry	0.034	0.001	0.222	9.495	-1.259	4.728	142.738
stMom	0.034	0.005	0.224	9.277	-1.227	4.728	137.292
ltMom	0.034	0.003	0.212	9.308	-1.204	4.127	135.521
value	-0.062	-0.030	0.400	-0.665	-6.265	7.363	37.510
NFA	0.003	0.001	0.033	2.548	-0.509	0.501	103.832
LTY	1.115	0.696	3.042	13.408	-9.031	105.370	353.198
TS	1.008	1.039	2.406	13.220	-29.453	89.950	377.986

set that comprises over 200 distinct baseline signals, thereby ensuring that the model accounts for both granular currency-level details and macroeconomic influences in a comprehensive and methodical manner.

4. Methodology

We aim to cross-sectionally predict currency excess returns, denoted as $E_t(r_{i,t+n})$. To achieve this, we employ machine learning models as functions of predictor variables, denoted as $g^\star$. The primary objective here is to explore the power of machine learning tools in predicting currency excess returns. One comparative approach involves identifying which model can maximize the out-of-sample explanatory power for realized returns denoted as $r_{i,t+n}$. For the inputs of the machine learning model, we use $z_{i,t}$ to represent the predictors. Thus, mathematically, our problem can be formulated as follows: to estimate the conditional asset pricing equation using machine

learning $\mathrm{E}_t\left(r_{i,t+n}\right) = g^\star\left(z_{i,t}\right)$. Here $g^\star\left(z_{i,t}\right)$ represents the linear/nonlinear transformation of the interactions of the predictors/covariates $z_{i,t}$ depending on the choice of the machine learning model.

Specifically, relating to the standard beta pricing representation of asset pricing, denoted as $E_t(r_{i,t+n}) = \beta_{i,t}\gamma_t$, we assume that the predictors allow for covariates between currency-specific characteristics and general conditions. Mathematically, we can represent the predictor as $z_{i,t} = m_t \times s_{i,t}$, where m_t represents general conditions common to all currencies, and $s_{i,t}$ represents currency-specific characteristics. We discuss the methods and applications of machine learning, rather than traditionally estimating β and γ. Thus, we do not directly estimate and evaluate β and γ. One significant advantage of machine learning is its ability to accommodate more flexible functional forms for $g^\star$.

4.1. Model specification

Ordinary least squares (OLS)

The regression is:

$$r_{it} = X_{it}\beta_{it} + \epsilon_{it},$$

where r_{it} is the excess return for individual currency pair i at time t (monthly return), X_{it} represents the predictors for the same currency pair at time t, β_{it} denotes the coefficients for individual i at time t, and ϵ_{it} represents the error term for individual i at time t. Here, we use pooled Ordinary Least Squares regression for estimation. Our objective function for estimation is to minimize the sum of squared residuals across all currencies and time periods. Mathematically, this is an optimization question:

$$\min_{\beta} \sum_{i=1}^{N}\sum_{t=1}^{T} \epsilon_{it}^2, \quad \text{where } \epsilon_{it} = r_{it} - X_{it}\beta_{it}.$$

For each individual i and time t, we get the β_{it} estimates by minimizing the squared residuals:

$$\beta_{it} = \arg\min \sum_{j=1}^{i\times t} (r_{it} - X_{it}\beta_{it})^2.$$

We use the default mean squared error (MSE) as our objective function for optimization. For the next step, we replace the default error function with the Huber loss function. The Huber loss function combines the benefits of both squared error and the absolute error. This make it less

sensitive to outliers. It is helpful for further investigation on the potential improvements in the model's predictive performance. Below, we provide the specific set-up for the Huber loss function, adapted for a pooled OLS setup:

$$L_\delta(r_{i,t}, \hat{r}_{i,t}) = \begin{cases} \frac{1}{2}(r_{i,t} - \hat{r}_{i,t})^2, & \text{if } |r_{i,t} - \hat{r}_{i,t}| \leq \delta, \\ \delta |r_{i,t} - \hat{r}_{i,t}| - \frac{\delta^2}{2}, & \text{if } |r_{i,t} - \hat{r}_{i,t}| > \delta. \end{cases}$$

$r_{i,t}$ is the actual return for unit i at time t, $\hat{r}_{i,t}$ refers to the predicted return for unit i at time t, δ is a threshold parameter that determines the transition point from quadratic to linear loss.

This set-up ensures residuals are penalized differently. Specifically, the smaller residuals are penalized heavily using the second moments, while the larger residuals are subject to linear penalties. This design is particularly effective in reducing the influence of outliers, thereby enhancing the model's robustness and overall predictive stability. The experiments provide insights into how different error functions impact the performance and generalizability of the pooled OLS model.

Neural networks (NN)

As neural nets are highly parameterised, it is easy to overfit. We use the regularization methods discussed in Gu et al. (2020), say learning rate shrinkage (incorporated in Adam solver) and early stoppings. Besides, another requirement from so many parameters is more data. Therefore, if one uses tiny dataset to have a taste of neural nets, it is very likely that it underperforms simpler models.

The architecture of a neural network—composed of an input layer, hidden layers, and an output layer—defines how the data flows and how the model extracts information from this data. The way the network extracts patterns from the data is influenced by its depth and the selection of activation functions, both of which affect its capacity to manage complex tasks. The fully connected nature of neural networks allows each neuron to influence the subsequent layers, enabling the learning of hierarchical patterns. This general architecture can be adapted for various tasks, such as classification, regression, and even more complex applications like image recognition or natural language processing. For illustration, Figure 1 shows a NN model with 5 layers fully connected .

In a neural network, the output is computed in a sequential, layer-wise manner, where each layer transforms its input into an output that serves as the input for the subsequent layer. Let us denote the input to the k-th layer as $\mathbf{h}_{k-1}$, and its output as $\mathbf{h}_k$. Mathematically, the computation

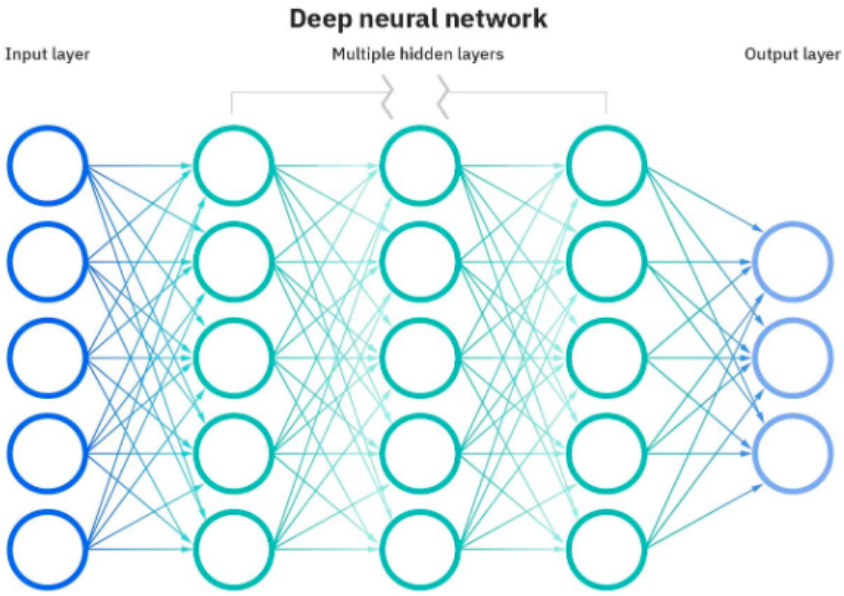


Figure 1: A deep neural network architecture (Pakkanen, 2021)

within the k-th layer can be expressed as (Pakkanen, 2021):

$$\mathbf{h}_k = \sigma(\mathbf{W}_k \mathbf{h}_{k-1} + \mathbf{b}_k),$$

where $\mathbf{W}_k$ is the weight matrix associated with the k-th layer, representing the trainable parameters that determine the strength and direction of the connections between neurons in the $(k-1)$-th and k-th layers; $\mathbf{b}_k$ is the bias vector for the k-th layer, another set of trainable parameters that allow the model to shift the activation function's response and prevent it from being constrained around zero; $\sigma(\cdot)$ is the activation function, which introduces non-linearity to the model, enabling the network to learn complex, non-linear patterns in the data. Without non-linear activation functions, the entire neural network would collapse to a linear model, regardless of the number of layers or neurons. Non-linear functions enable the model to approximate complicated mappings from inputs to outputs, making neural networks powerful for tasks such as image recognition, natural language processing, and more. For example: the ReLU function $\sigma(x) = \max(0, x)$ is widely used due to its simplicity and efficiency in mitigating the vanishing gradient problem during training; the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ maps inputs to a range between 0 and 1, which is useful in probabilistic interpretation tasks; the tanh function $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ maps inputs to a range between -1 and 1,

centering outputs around zero. The forward computation proceeds layer by layer, starting from the input layer and propagating through each hidden layer. The final output of the network, denoted as $\hat{y}$, is computed in the last layer (the L-th layer) as:

$$\hat{y} = \mathbf{W}_L \mathbf{h}_{L-1} + \mathbf{b}_L,$$

where $\mathbf{W}_L$ and $\mathbf{b}_L$ are the weight matrix and bias vector for the final layer, and $\mathbf{h}_{L-1}$ is the output of the last hidden layer.

In this context, the weights $\mathbf{W}_k$ and biases $\mathbf{b}_k$ are the parameters of the neural network. These are learned during training by minimizing a predefined loss function, such as mean squared error, depending on the nature of the task (e.g., regression or classification). Optimization algorithms, such as stochastic gradient descent (SGD) or its variants (e.g., Adam), are employed to iteratively adjust $\mathbf{W}_k$ and $\mathbf{b}_k$ in the direction that reduces the loss. In contrast, the hyperparameters refer to choices made before training that are not directly learned from the data. These include the layer count (L), neuron distribution per layer, activation function $\sigma(\cdot)$, learning rate, and regularization methods (e.g., dropout).

From the perspective of our predictive framework, we can interpret this neural network structure in relation to the asset pricing representation introduced in Section 4. Specifically, the first layer's input $\mathbf{h}_1$ corresponds directly to our predictor variables $z_{i,t}$, which are constructed as the product of market-wide economic conditions m_t and currency-specific characteristics $s_{i,t}$,

$$z_{i,t} = m_t \times s_{i,t}.$$

In this context, $\mathbf{h}_1$ is the fundamental set of inputs capturing both global and currency-specific influences. As the data propagates through multiple layers, each transformation $\mathbf{h}_k = \sigma(\mathbf{W}_k \mathbf{h}_{k-1} + \mathbf{b}_k)$ represents a combination of linear transformations (determined by $\mathbf{W}_k$ and $\mathbf{b}_k$) and non-linear activations (through $\sigma(\cdot)$). This recursive composition allows the model to extract complex interactions and higher-order relationships from the predictors. Finally, the last layer's output $\mathbf{h}_n$ corresponds to our function $g^*(z_{i,t})$, which represents the predictive mapping of our machine learning model:

$$E_t(r_{i,t+n}) = g^*(z_{i,t}).$$

Thus, we can conceptualize the entire neural network as an iterative refinement process, where each layer progressively transforms the predictor variables to capture intricate structures and dependencies, ultimately leading to the best estimation of conditional expected returns. The presence of multiple layers enables the model to approximate non-linear dependencies in asset pricing. The asset return prediction $E_t(r_{i,t+n})$ is obtained through a deep hierarchical structure refining the interactions between $z_{i,t}$ and the factor loadings embedded within the network's weight parameters.

Dimension reduction: Principal Components regression (PCR) and Partial least squares (PLS)

Our setting is characterized by its high dimensionality of the predictor set, which is especially prevalent in asset pricing (Nagel, 2021). In particular, we deal with a large number of predictors, while the test assets' cross-sectional dimension—in this case, currencies—is modest. There may be a high degree of correlation between variables, causing the model to be unstable. Multicollinearity can make the estimation of regression coefficients inaccurate or even uninterpretable. Furthermore, the risk of overfitting arises because of the small number of data observations in comparison to the size of the predictor set. When a model gets overly complicated, it overfits and captures noise instead of important signals, which results in poor generalization on fresh data. High-dimensional data leads to a dramatic increase in computational costs, especially in matrix operations. For example, computing the inverse or eigenvalue decomposition of a high-dimensional matrix can be very time-consuming.

PCR (Massy, 1965) can be understood as a combination of two parts as below. In the first part, we apply principal component analysis to extract the most significant componets in the predictor set. In the second part, we regress returns on these principal components. In the first step, we measure the significance of factors mainly by how well they can capture variance in the whole predictor set. Compared with OLS, this method simplifies the model while retaining the linear relationship, and reduces the risk of model instability problems such as overfitting. Compared with neural network, this method also similarly puts different weights on the predictors to retain the most effective information, but here we still only consider the linear relationship between the predictor and the returns.

Another dimension reduction method is PLS (Wold, 1966). The main difference between it and PCR is that the method of weighting predictors is different. PCR's weighting on principle

components is based on the predictor set itself, which reduces dimensionality while retaining most variance in the predictor set. But the drawback is that some high-variance components may be irrelevant for predicting the returns. Instead, PLS extends PCR by selecting components that maximize the covariance between the predictor set and the returns, ensuring that the reduced feature space is optimized for prediction.

Both PCR and PLS provide solutions to high-dimensional regression problems through dimensionality reduction. The focus of PCR is to maximize variance of the predicto set while PLS directly optimizes prediction. The choice between PCR and PLS depends on whether feature variance (PCR) or predictive power (PLS) is more important in a given application. PLS may be more efficient than PCR when the predictor set and the returns are strongly correlated.

Tree-based models: Random forest (RF) and Gradient Boosted Regression Trees (GBRT)

Tree-based models are a class of machine learning algorithms valued for their interpretability, flexibility, and capacity to model nonlinear relationships. These models recursively partition the feature space into regions associated with leaf nodes, using splitting rules that optimize target homogeneity. The fundamental unit, the decision tree, segments data via feature-based rules and assigns predictions at the leaves. As shown in Figure 2, a decision tree comprises nodes, branches, and leaves: nodes represent decision points, branches indicate outcomes, and leaves provide final predictions—either class labels or continuous values. Tree construction involves selecting features and split points that maximize the purity of resulting subsets, commonly measured by Gini impurity for classification and mean squared error (MSE) for regression (Breiman, 2001). Although decision trees are intuitive and powerful, they are prone to overfitting. Despite their interpretability, decision trees are prone to overfitting, particularly when excessively deep. This drawback motivates ensemble approaches such as Random Forests (RF) and Gradient Boosted Regression Trees (GBRT), which aggregate multiple trees to enhance predictive accuracy and generalization. Compared to neural networks, tree-based models provide competitive nonlinear modeling with greater interpretability.

Random Forest (RF) (Breiman, 2001) is an ensemble learning method. It builds up multiple decision trees during training and averages their predictions to improve accuracy and reduce overfitting. The key idea is to introduce randomness through bootstrapped sampling of training data

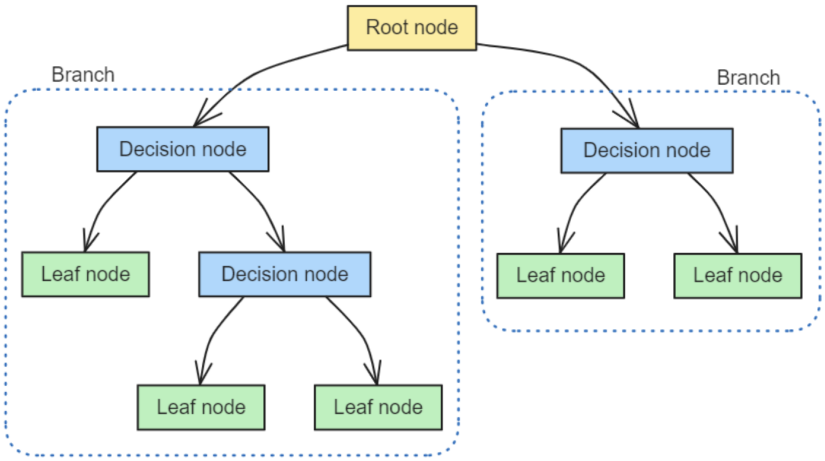


Figure 2: A tree-based model (Andrés, 2023)

(Bagging) and random feature selection at each split. This decorrelates individual trees and leads to a robust and stable model.

Gradient Boosted Regression Trees (GBRT) (Friedman, 2001) is an ensemble learning approach based on decision trees. However, it differs fundamentally from Random Forest. Instead of training trees separately, GBRT sets up trees in a sequential manner, with each tree correcting the mistakes of its predecessors by fitting the preceding trees' residuals. Although iterative optimization enhances prediction accuracy, it can lead to overfitting if not appropriately regularized.

Gradient Boosting Regression Tree (GBRT) and Random Forest (RF) differ in performance characteristics and training schemes. Random Forest adopts self aggregation (bagging), allowing each tree to be trained independently in parallel, while GBRT is trained sequentially, where each tree attempts to correct the errors of the previous tree (i.e. boosting). Due to this fundamental difference, random forests are typically more robust to noise and capable of handling high-dimensional sparse information. GBRT can provide higher prediction accuracy with reasonable adjustments, but it is also more prone to overfitting. Although GBRT performs particularly well on structured data with clear patterns, careful adjustment of hyperparameters such as tree depth and learning rate is necessary to avoid overfitting. Although random forests provide a stable and well generalized method, the iterative loss function minimization mechanism of GBRT makes it a powerful tool for optimizing prediction accuracy.

Previous studies have primarily employed linear models such as ordinary least squares (OLS) and robust variants (e.g., OLS with Huber loss) to address outliers. Dimensionality reduction

techniques like principal component regression (PCR) and partial least squares (PLS) are used to mitigate multicollinearity and high dimensionality. While these models are well-suited for linear relationships and offer interpretability through coefficient estimates, they often fail to capture non-linear interactions present in real-world data, limiting their predictive performance. In contrast, neural networks (NN) are capable of learning highly flexible, nonlinear mappings and are effective in extracting deep feature representations, particularly for unstructured data. However, their lack of interpretability—the so-called “black box" problem—poses challenges in domains such as finance and healthcare, where transparency is essential (Yuan et al., 2024). Tree-based models offer a middle ground by capturing nonlinear patterns while retaining a degree of interpretability (Hastie, 2009). Decision trees, the basis for Random Forests (RF) and Gradient Boosted Regression Trees (GBRT), generate rule-based structures that are easily visualized and allow for feature importance analysis (Breiman, 2001). Unlike linear models, they automatically account for interactions, and unlike neural networks, they perform well on heterogeneous, structured data with modest data and computational requirements (James et al., 2013). In sum, while linear models remain important for inference and neural networks excel with complex, high-dimensional inputs, tree-based methods provide a robust and interpretable alternative for predictive modeling on structured data.

4.2. Model evaluation

Correlation and prediction error

Model performance is primarily evaluated using the mean squared error (MSE), defined as the squared difference between actual and predicted returns. Minimizing MSE aligns with the objective of improving predictive accuracy by reducing deviations from observed data. Additionally, we report the correlation coefficient between predictions and observations as a bounded metric to quantify the strength of their linear relationship.

By minimizing the sqaured error term, we aim to reduce the model's prediction inaccuracies and enhance its ability to generalize effectively across both in-sample and out-of-sample data. As seen from the formula, in our scenario, the objective of minimising the squared error term is equivalent to maximising R^2.

Diebold-Mariano (DM) test implementation

In order to compare and analyze the prediction models and evaluate whether there is a significant difference in prediction accuracy between the two models, we refer to Gu et al. (2020) to implement

the DM test. This test is based on the evaluation of prediction error over time: $\mathrm{e}_{i,t+1}^{(1)} = r_{i,t+1} - \hat{r}_{i,t+1}^{(1)}$, $e_{i,t+1}^{(2)} = r_{i,t+1} - \hat{r}_{i,t+1}^{(2)}$, where $r_{i,t+1}$ is the realized target value, currency return, while $\hat{r}_{i,t+1}^{(1)}$ and $\hat{r}_{i,t+1}^{(2)}$ are the predicted values generated by Model 1 and Model 2, respectively. The test is constructed around a loss function, here chosen to be the squared error, $L(e) = e^2$, to quantify the accuracy of each model's predictions. A key adaptation in the present study involves focusing not on individual prediction errors but on the cross-sectional average loss differential at each time point. The cross-sectional mean loss differential is computed as:

$$d_{12,t+1} = \frac{1}{n_{3,t+1}} \sum_{i=1}^{n_{3,t+1}} \left[L(e_{i,t+1}^{(1)}) - L(e_{i,t+1}^{(2)}) \right],$$

where $n_{3,t+1}$ denotes the number of observations in the cross-sectional sample at time $t+1$.

The test statistic, DM_{12}, is calculated using the time series of cross-sectional loss differentials and is defined as:

$$\mathrm{DM}_{12} = \frac{\bar{d}_{12}}{\hat{\sigma}_{d_{12}}},$$

where $\bar{d}_{12}$ is the time-averaged mean loss differential given by:

$$\bar{d}_{12} = \frac{1}{T} \sum_{t=1}^{T} d_{12,t},$$

and $\hat{\sigma}_{d_{12}}$ is its standard error, estimated using the Newey-West approach to account for potential autocorrelation. Specifically, the Newey-West estimator incorporates both contemporaneous variance and autocovariances (Newey and West, 1987), and is expressed as:

$$\hat{\sigma}_{d_{12}}^2 = \hat{\gamma}(0) + 2 \sum_{k=1}^{h-1} \hat{\gamma}(k),$$

where the autocovariance at lag k is given by:

$$\hat{\gamma}(k) = \frac{1}{T} \sum_{t=k+1}^{T} \left(d_{12,t} - \bar{d}_{12} \right) \left(d_{12,t-k} - \bar{d}_{12} \right),$$

and h is the bandwidth parameter controlling the maximum lag length.

Under the null hypothesis H_0, which asserts that the two models have equivalent predictive accuracy on average ($\mathbb{E}[d_{12,t}] = 0$), the test statistic DM_{12} asymptotically follows a standard normal

distribution, i.e., $DM_{12} \sim \mathcal{N}(0,1)$. The null hypothesis is rejected at a significance level α if the absolute value of the test statistic exceeds the critical value $Z_{\alpha/2}$, where $Z_{\alpha/2}$ represents the $(1-\alpha/2)$-quantile of the standard normal distribution.

This cross-sectional adaptation of the Diebold-Mariano test is particularly advantageous in financial applications where prediction errors exhibit strong dependencies across assets due to shared economic and market factors. By focusing on aggregated cross-sectional measures of model performance, this approach reduces the influence of idiosyncratic noise while capturing the broader trends in prediction accuracy. Newey-West standard errors are what we actually utilize. This ensures the test's validity even when autocorrelation is present.

5. Empirical results

Based on the temporal order, we divide the dataset into three separate subsets: test, validation, and training. The model is developed using the training subset. Its parameters are adjusted using the validation subset. The test subset is subjected to the trained model, which is intact following training. This structure serves to ensure a thorough evaluation of the model's generalization capacity by assessing its predictive performance on data that hasn't been seen before.

5.1. Correlation

Table 2 reports the correlation coefficients between predicted and actual values for each model over a one-month prediction horizon, covering both in-sample and out-of-sample periods. As a standardized and interpretable metric, correlation measures the linear alignment between forecasts and realized outcomes. While it does not capture nonlinearity, it remains a widely used criterion for model evaluation. Among the models considered, OLS with Huber and shallow neural networks exhibit relatively strong correlations in both in-sample and out-of-sample settings. Table 2 shows a weaker overall correlation for long-term forecasts (e.g., 12 months) than for short-term predictions (e.g., 1 month).

Tree-based models such as Random Forest (RF) and Gradient Boosted Regression Trees (GBRT) exhibit strong in-sample performance, with GBRT achieving correlations as high as 0.9. However, their out-of-sample correlations are substantially lower, indicating limited generalization and potential overfitting. GBRT's boosting framework allows for highly accurate in-sample fitting by sequentially reducing residuals, but it is prone to overfitting, especially with small datasets. In

contrast, RF leverages the bagging strategy—training multiple independent trees on bootstrapped samples—which improves robustness and typically yields higher out-of-sample correlation than GBRT.

Linear models (e.g., OLS, OLS with Huber loss, PLS, and PCR) generally perform well due to the approximately linear relationships present in many financial variables, and they remain widely used in asset pricing and macroeconomic forecasting. For example, the linear regression framework underpins the Fama-French three-factor model Fama and French (1993) and the Carhart four-factor model (Carhart, 1997). While OLS shows high in-sample correlation, its out-of-sample performance is weaker, reflecting limited generalizability. PLS and PCR, which reduce dimensionality by extracting latent components, are more robust in high-dimensional settings and yield superior out-of-sample correlations relative to OLS, mitigating overfitting to some extent.

In particular, the out-of-sample correlation of shallow networks is even negative, but the performance of somewhat deeper networks has improved but is still inferior to RF. Neural network models exhibit more instability. There might be a number of reasons for this. The model may not be able to discover consistent patterns because, first of all, neural networks typically need a lot of data to train efficiently, and financial market data is frequently sparse and extremely noisy. Second, the gradient descent technique is used in the neural network's optimization process. When dealing with high-noise input, the gradient update may become unstable or even fall into a local optimum. Furthermore, the choice of neural network hyperparameters significantly affects the outcome. The model's ultimate performance may be impacted by the number of layers, activation functions, regularization techniques, etc. Underfitting or overfitting may occur from improper hyperparameter adjustment.

It should be noted that as a non-linear model as well the tree-based models also have a very high correlation. One explanation might be that tree-based models are more robust to noise and more adaptable when handling high-dimensional, nonlinear characteristics. Specifically, RF chooses features and samples at random, which increases its robustness in out-of-sample assessment. In contrast, GBRT's boosting process allows it to learn more intricate patterns, but it also makes it more susceptible to overfitting. In reference to this field of study, Gu et al. (2020) examined how well several machine learning models performed in asset pricing. In several instances, particularly when forecasting asset returns, they discovered that tree-based techniques outperformed neural

networks in terms of stability.

Finding a balance between neural networks and tree-based models may be necessary from the standpoint of practical application if the objective is to get more reliable out-of-sample prediction performance. To lessen overfitting, we may, for instance, consider increasing regularization throughout GBRT training phase or lowering the tree's maximum depth. In order to increase the neural network's capacity for generalization, one may also attempt to better tune its hyperparameters, such as by changing the batch size, learning rate, dropout rate, etc. Furthermore, as neural networks demand more data to train, data augmentation techniques like building more features or generating more samples using rolling window approaches might be used in the future to boost the model's performance. In the end, selecting the best model necessitates careful evaluation of the model's stability, interpretability, and resilience under various market situations in addition to in-sample or out-of-sample correlation.

Table 2: **Monthly in-sample and out-of-sample currency-level prediction performance (Correlation)**

This table presents the monthly prediction performance at the currency level, comparing in-sample and out-of-sample correlation values. The data is divided into training, validation, and testing samples. The model is trained and validated on the training and validation data, respectively, and then applied to the test data without further adjustment. The evaluation includes five models: Ordinary Least Squares (OLS), OLS with Huber loss (OLSH), neural networks (NN) with increasing complexity, from one to eight hidden layers, the dimension reduction linear models including Partial Least Squares (PLS) and Principal Component Regression (PCR), as well as the tree-based models including Random Forest (RF) and Gradient Boosted Regression Trees (GBRT). Correlation, as a bounded measure ranging from -1 to +1, quantifies the strength and direction of the linear relationship between predicted and actual values. A correlation of +1 signifies a perfect positive linear relationship, where the model's predictions move in exact proportion to the observed values. Conversely, a correlation of -1 indicates a perfect negative linear relationship, meaning the model systematically predicts the opposite of the actual outcomes. A correlation close to zero suggests little to no linear relationship, implying that the model's predictions have limited explanatory power for the observed data.

Prediction horizon: one month		
Model	In-Sample	Out-of-Sample
OLS	0.977	-0.142
OLS+Huber (OLSH)	0.973	0.926
Neural Network with One Hidden Layer	0.759	0.298
Neural Network with Two Hidden Layers	0.807	0.888
Neural Network with Three Hidden Layers	0.437	0.253
Neural Network with Four Hidden Layers	0.963	0.802
Neural Network with Five Hidden Layers	0.270	0.063
Neural Network with Six Hidden Layers	0.938	0.481
Neural Network with Seven Hidden Layers	0.958	0.236
Neural Network with Eight Hidden Layers	0.781	0.882
Partial Least Squares (PLS)	0.977	0.902
Principal Component Regression (PCR)	0.932	0.903
Random Forest (RF)	0.974	0.907
Gradient Boosted Regression Trees (GBRT)	0.993	0.509
Prediction horizon: twelve months		
Model	In-Sample	Out-of-Sample
OLS	0.770	-0.472
OLS+Huber (OLSH)	0.708	0.718
Neural Network with One Hidden Layer	0.089	-0.165
Neural Network with Two Hidden Layers	0.635	-0.015
Neural Network with Three Hidden Layers	0.570	0.458
Neural Network with Four Hidden Layers	0.661	0.564
Neural Network with Five Hidden Layers	0.385	-0.167
Neural Network with Six Hidden Layers	0.617	0.328
Neural Network with Seven Hidden Layers	0.699	0.427
Neural Network with Eight Hidden Layers	0.667	-0.052
Partial Least Squares (PLS)	0.656	0.667
Principal Component Regression (PCR)	0.646	0.652
Random Forest (RF)	0.848	0.668
Gradient Boosted Regression Trees (GBRT)	0.962	0.073

5.2. Prediction error

In Table 3, we use Mean Squared Error (MSE) as a statistic to help assess prediction accuracy and examine the monthly in-sample and out-of-sample forecast performance of several forecast models. This research shows the benefits and drawbacks of various optimization techniques in addition to assisting in understanding the generalization potential of various models. We discovered the followings from the experimental data that need discussion.

The model's low out-of-sample generalization performance can be attributed to a number of causes. Due to the substantial noise and non-stationarity of financial time series data, many models may fit training data well but perform badly on new data sets. This implies that overfitting, or the model's overlearning of particular patterns in the data during training, may be the cause of the model's instability in subsequent data. Furthermore, certain particular variables may lose their predictive power on out-of-sample data due to dynamic changes in the market environment, which calls into question the model's stability in real-world applications.

Second, forecasts that are twelve months in length perform noticeably worse than those that are one month in length. Financial forecasting tasks frequently exhibit this tendency, which might be caused by a number of factors: First, financial markets can be predicted in the short term, but it is more challenging to anticipate in the long run. While long-term models must contend with more macroeconomic shifts, uncertainties, and structural disruptions, which increases the likelihood of long-term forecast errors, short-term models can use current market data to spot short-term trends. Furthermore, conventional modeling techniques often work better over shorter time periods, but noise and external shocks can lead to model failure over longer time periods.

Regarding model type, we found that out-of-sample, nonlinear models—such as neural networks and tree-based techniques—generally perform better than linear models—such as OLS regression and dimensionality reduction techniques. The benefits of nonlinear models in managing intricate data structures and nonlinear interactions might be the cause of this development. The connections between variables in financial markets, on the other hand, frequently exhibit nonlinear properties, whereas linear models typically assume linear interactions between variables. By using intricate functional mapping relationships, neural networks and tree-based techniques can better anticipate outcomes by capturing these nonlinear features. Though in-sample performance may not be evident, this advantage is mostly seen in out-of-sample predictions, suggesting that these models may exhibit

more robust adaptive skills during training but may still exhibit unstable performance in the event of overfitting.

The OLS regression model's Huber error function, also known as Huber loss, handles outliers better than other linear models. In order to lessen the effect of outliers on model parameter estimation, the Huber loss function processes the error piecewise, making it comparable to the mean square error (MSE) when the error is little and to the absolute error (MAE) when the error is big. As a result, even in the face of dramatic market fluctuations, the model remains resilient and retains its strong prediction powers. The contribution of dimension reduction approaches to prediction performance is limited. The model may not be able to fully exploit all possible prediction signals as a result of the significant information loss that occurs throughout the dimensionality reduction procedure. Furthermore, while the interaction between variables in financial markets can be quite complicated, dimensionality reduction techniques often presume that the data has a low-dimensional structure. while a result, direct dimensionality reduction may lose important information, which could diminish forecast accuracy.

Subsequent examination of the nonlinear model revealed that the neural network model's prediction accuracy did not systematically increase as the number of layers increased. This demonstrates that in a certain data setting, more sophisticated models may result in overfitting rather than performance gains. While deep neural networks can improve the model's expressiveness, they may also learn the noise in the data rather than valuable prediction signals if there is insufficient training data or insufficient effective information in the data. Furthermore, neural networks may be very dependent on training data in out-of-sample prediction tasks, which might lead to inadequate generalization skills. To prevent overfitting issues, complexity and data size must be considered while selecting a neural network structure.

Likewise, there is no discernible pattern of variation in prediction performance across neural networks and tree-based models. This could have to do with how the two models differ and how they are similar. By using a splitting rule-based decision-making process that automatically chooses important characteristics and manages nonlinear interactions, tree models—such as random forests and gradient boosted trees—model nonlinear connections. For neural networks to identify patterns in data, weight optimization is essential. The two may perform similarly in real-world applications, despite their theoretically larger expressive capacities. Furthermore, when working with

high-dimensional data, tree models could be more stable, whereas neural networks need a lot of regularization and parameter adjustment to avoid overfitting. As a result, both neural networks and tree-based approaches have benefits, and how well they perform varies on the hyperparameter settings and the properties of the data.

In conclusion, this study's experimental findings highlight a few critical elements that influence the model's capacity for prediction. First, out-of-sample generalization skills, particularly long-term prediction errors, are often poor due to the significant noise and non-stationarity of financial markets. Second, when it comes to forecasting out-of-sample, nonlinear models perform better than linear models, suggesting that there could be significant nonlinear correlations in financial markets. In addition, the OLS model's resilience is increased by using the Huber loss function, although dimensionality reduction techniques have a limited impact. Lastly, there is no discernible improvement in prediction accuracy from neural network or tree-based approaches, suggesting that rather than depending only on more intricate model structures, the model selection procedure should be tailored to certain data features. To increase the model's resilience and capacity for generalization, future studies might investigate various regularization techniques, feature engineering approaches, and ensemble learning techniques.

Table 3: **Monthly in-sample and out-of-sample currency-level prediction performance (mean squared error)**

This table presents the monthly prediction performance at the currency level, comparing in-sample and out-of-sample, measured by the mean squared error. In this scheme, the data is divided into training, validation, and testing samples. The model is trained and validated on the training and validation data, respectively, and then applied to the test data without further adjustment. The evaluation includes five models: Ordinary Least Squares (OLS), which minimizes squared errors but shows poor out-of-sample performance; OLS combined with the Huber loss function (OLS+Huber), which is more robust to outliers and performs better out-of-sample; neural networks with increasing complexity, featuring one to eight hidden layers; the dimension reduction linear models including Partial Least Squares (PLS) and Principal Component Regression (PCR); and the tree-based models including Random Forest (RF) and Gradient Boosted Regression Trees (GBRT). A smaller mean squared error indicates improved predictive performance. Year of 2010 is the split between the in-sample and out-of-sample.

Prediction horizon: one month		
Model	In-sample	Out-of-sample
OLS	0.004	305.820
OLS+Huber (OLSH)	0.005	4.838
Neural Network with One Hidden Layer (NN1)	0.161	1.990
Neural Network with Two Hidden Layers (NN2)	0.054	1.063
Neural Network with Three Hidden Layers (NN3)	0.080	0.941
Neural Network with Four Hidden Layers (NN4)	0.011	11.331
Neural Network with Five Hidden Layers (NN5)	0.251	3.108
Neural Network with Six Hidden Layers (NN6)	0.085	0.601
Neural Network with Seven Hidden Layers (NN7)	0.051	0.510
Neural Network with Eight Hidden Layers (NN8)	0.045	12.497
Partial Least Squares (PLS)	0.004	16.787
Principal Component Regression (PCR)	0.012	17.265
Random Forest (RF)	0.005	11.058
Gradient Boosted Regression Trees (GBRT)	0.001	6.382
Prediction horizon: twelve months		
	In-sample	Out-of-sample
OLS	0.038	1237.424
OLS+Huber (OLSH)	0.048	4.039
Neural Network with One Hidden Layer (NN1)	0.658	11.591
Neural Network with Two Hidden Layers (NN2)	0.061	0.745
Neural Network with Three Hidden Layers (NN3)	0.125	4.947
Neural Network with Four Hidden Layers (NN4)	0.110	7.072
Neural Network with Five Hidden Layers (NN5)	0.084	0.732
Neural Network with Six Hidden Layers (NN6)	0.061	1.842
Neural Network with Seven Hidden Layers (NN7)	0.071	1.534
Neural Network with Eight Hidden Layers (NN8)	0.081	0.260
Partial Least Squares (PLS)	0.053	9.607
Principal Component Regression (PCR)	0.054	8.716
Random Forest (RF)	0.028	9.713
Gradient Boosted Regression Trees (GBRT)	0.007	1.052

5.3. Diebold-Mariano (DM) test

From Table 4, the findings of this study's Diebold-Mariano (DM) test demonstrate that neural networks (NN) outperform linear models and random walk (RW) models statistically substantially across all prediction horizons. The neural network's out-of-sample performance is much better than that of the random walk (RW), in addition to being superior than the OLS linear regression model in terms of predicting ability. Research on predictability of the exchange rate over time has long held the view that RW, a straightforward model without any parameters, is even superior to conventional econometric models.

The random walk (RW) is still a more reliable baseline model, even if the neural network model did the best in this investigation. RW fared better than the linear regression model. For instance, RW is much better than OLS-Huber and ranks second only to NN7 in the one-month prediction test, while it is significantly better than OLS-Huber and ranks second only to NN8 in the twelve-month prediction job. This demonstrates that RW is still a powerful comparison benchmark, even though advanced machine learning techniques can offer additional predictive benefits.

The influence of the number of neural network layers on the predictive capacity was also demonstrated by the DM test results. Overall, as the number of hidden layers in the neural network rises, so does the prediction accuracy. NN7 with seven hidden layers performed best in the one-month prediction task, and NN8 with eight hidden layers performed best in the twelve-month prediction challenge. This implies that a neural network's forecasting capacity is frequently improved by deepening it, since deeper networks are better able to extract latent characteristics from the market and learn more intricate nonlinear patterns. It should be highlighted, nevertheless, that although adding more layers might enhance the model's expressive capabilities, going overboard with the network depth can raise computational expenses and result in overfitting issues when there is not enough data. Instead of mindlessly adding more layers, actual applications require that the neural network's depth be modified based on the particular prediction goal.

Furthermore, the DM test results demonstrate that the more successful conventional models vary depending on the prediction horizon. OLS-Huber is the model that outperforms NN7 and RW in the one-month prediction challenge. By adding the Huber error function, the model becomes more resilient to outliers and outperforms conventional OLS in short-term forecasting. Gradient Boosting Tree (GBRT), rather than OLS-Huber, is the model that performs better in the twelve-

month prediction challenge. The benefits of tree-based models in managing high-dimensional data and long-term nonlinear patterns may be connected to GBRT's better long-term forecasting ability. It demonstrates that rather than merely assuming that a particular model performs optimally across all forecasting jobs, model selection in financial forecasting activities must be optimized in tandem with the forecast's time span.

When taken together, the study's DM test findings demonstrate that the neural network model beats the random walk and linear regression models by a wide margin and performs ideally in all prediction tasks. This research suggests that deep learning techniques may have more application potential in financial prediction, which somewhat contradicts conventional conclusions in the literature. However, the random walk model continues to be a strong baseline, surpassing linear regression on a number of tasks and coming in second only behind neural networks. Furthermore, as the number of layers in neural networks rises, their predictive power often improves as well. However, in order to avoid overfitting, the complexity must be kept under control. The best conventional models vary depending on the predicted timeframe. In short-term prediction, the OLS-Huber error function performs better, while in long-term prediction, the gradient boosting tree (GBRT) offers more benefits. These findings suggest that choosing a model cannot be done just by relying on one technique; rather, it must be improved in tandem with certain prediction tasks. However, there is always room for improvement in deep learning techniques, particularly in terms of how to better explain their predictive power, enhance generalization performance, and lower computing costs.

Table 4: **Comparison of monthly out-of-sample prediction using Diebold-Mariano tests**

This table presents the results of pairwise Diebold-Mariano (DM) test statistics comparing the out-of-sample predictive performance of currency-level forecasting models. We cover the linear models (OLS, OH, PLS, PCR), neural networks (NN), and tree-based models (RF, GBRT). Each cell in the table shows the DM test statistic calculated based on the null hypothesis that the prediction accuracy (measured by mean squared error) of the row model is equal to that of the column model. A negative test statistic indicates that the column model exhibits lower mean squared error and thus outperforms the row model in prediction accuracy. It should be noted that all entries in the table are statistically significant at the 1% level, providing robust evidence to reject the null hypothesis.

Prediction horizon: one month

	OLS	OH	NN1	NN2	NN3	NN4	NN5	NN6	NN7	NN8	RW	PLS	PCR	RF	GBRT
OLS															
OH	110.529														
NN1	111.732	131.802													
NN2	111.915	458.658	47.189												
NN3	112.047	314.771	50.338	14.063											
NN4	108.251	-421.581	-560.319	-618.980	-548.148										
NN5	112.758	9.279	-6.293	-11.057	-11.795	45.643									
NN6	112.073	560.430	68.446	169.375	35.279	622.805	13.533								
NN7	112.117	510.090	70.515	180.878	46.767	592.591	14.079	31.881							
NN8	107.755	-716.324	-415.780	-771.961	-711.725	-65.741	-50.348	-784.090	-803.396						
RW	111.924	168.174	33.871	2.413	-3.091	386.856	11.251	-19.084	-23.182	443.068					
PLS	106.153	-987.048	-533.418	-866.245	-815.889	-271.681	-72.748	-911.933	-895.971	-311.563	-564.561				
PCR	105.927	-1244.014	-645.902	-1067.612	-897.877	-419.061	-76.107	-1119.432	-1049.792	-338.824	-622.056	-36.961			
RF	108.239	-888.163	-422.316	-1033.643	-794.797	19.518	-43.001	-1034.906	-1021.469	195.709	-429.803	408.263	578.253		
GBRT	109.934	-107.618	-193.234	-433.159	-342.050	259.329	-17.673	-457.364	-459.290	358.207	-216.808	490.318	604.009	369.411	

Prediction horizon: twelve months

	OLS	OH	NN1	NN2	NN3	NN4	NN5	NN6	NN7	NN8	RW	PLS	PCR	RF	GBRT
OLS															
OH	48.099														
NN1	47.791	-156.034													
NN2	48.231	32.461	86.744												
NN3	48.065	-66.517	117.467	-45.748											
NN4	47.977	-239.051	85.230	-67.450	-183.700										
NN5	48.240	321.802	222.320	0.130	240.333	342.890									
NN6	48.189	216.811	186.067	-11.152	272.438	461.066	-92.521								
NN7	48.202	351.743	210.869	-7.815	222.391	401.714	-107.960	31.028							
NN8	48.261	296.418	225.705	4.718	251.562	347.415	54.364	116.952	118.377						
RW	48.228	125.192	199.330	-2.468	143.270	214.979	-11.897	33.872	22.566	-30.983					
PLS	47.882	-804.070	40.800	-86.462	-326.390	-187.675	-810.698	-716.635	-886.494	-659.025	-345.359				
PCR	47.919	-695.873	57.315	-77.639	-273.078	-120.291	-768.385	-684.602	-768.410	-630.807	-312.716	256.320			
RF	47.875	-688.008	37.422	-87.169	-307.935	-170.573	-636.354	-610.994	-670.848	-589.393	-330.219	-11.141	-108.792		
GBRT	48.227	315.541	210.194	-3.023	251.583	350.192	-87.603	73.150	64.099	-90.857	-2.138	854.281	842.061	656.384	

6. Economic performance

In the next stage of the analysis, we construct a new set of portfolios specifically designed to capitalize on the return forecasts generated by machine learning models. At the end of each month, we compute one-month-ahead out-of-sample return predictions for each forecasting method. We then use these forecasts to divide equities into five quintiles according to the predicted returns that each model suggests. Since our forecasting techniques are trained to minimize equally weighted prediction errors, we use equal-weighting to create portfolios in order to ensure consistency with our statistical framework (Gu et al., 2020). This decision eliminates the need for further portfolio optimization considerations while enabling a more straightforward evaluation of forecast quality. Lastly, we employ a zero-net-investment strategy, which involves rebalancing the portfolio at the beginning of each month and taking long positions in stocks within the quintile with the highest anticipated return (quintile 5) and short positions within the quintile with the lowest predicted return (quintile 1). The performance metrics of these portfolios are shown. It appears that some machine learning models produce predictions that are negatively linked with realized returns since the realized returns do not always show a strictly monotonic connection with the projected returns from each model. It suggests that creating an ideal trading strategy may require more than just forecast rankings.

First, the best models for various prediction periods differ when viewed from the standpoint of economic value of predictability across models. With the greatest annualized Sharpe ratio of 1.103 under the one-month prediction horizon, NN7 demonstrate higher economic value in short-term trading. NN3 outperformed all other models with an annualized Sharpe ratio of 1.476 over the course of the twelve-month prediction horizon. It demonstrates that the best neural network architectures for short-term and long-term predictions differ significantly. More sophisticated neural networks (like NN7) may be better equipped to identify patterns in short-term market noise, which might lead to larger short-term trading profits. While relatively shallow neural networks (like NN3) are better able to extract long-term trend information and hence perform better under longer prediction periods, too sophisticated networks may overfit in long-term prediction.

Nevertheless, one consistent finding emerges: neural network-based models tend to outperform their linear counterparts, reaffirming their relative strength in capturing complex return dynamics. Neural network-based models consistently outperform linear models, despite the fact that the ideal

number of neural network layers changes depending on the prediction period. The neural network approach is always superior than linear regression (OLS, OLSH) and dimensionality reduction techniques like principle component regression (PCR) and partial least squares regression (PLS), regardless of the forecast period, which might be one month or twelve months. Neural networks can also occasionally perform better than tree-based models (such Random Forest RF and Gradient Boosted Tree GBRT). This result aligns with the structural features of neural networks. Neural networks are better able to uncover possible patterns in complicated market situations and capture the nonlinear aspects of asset returns than classic linear models. Neural networks can therefore provide some prediction benefits even when there is a high level of noise in the return data. Financial market return data may exhibit significant nonlinearities, making it challenging for linear models to accurately capture these intricate connections. As a result, linear models have a low predictive ability. Second, there could still be a lot of space for optimization of the characteristic variables in this study even if tree-based models (RF, GBRT) can theoretically adjust to some nonlinear features. For instance, feature engineering might not be able to adequately represent the market's structural shifts, which would restrict the tree model's capacity for generalization. Lastly, tree models may have a tendency to overfit the training data and be unable to successfully extract long-term trend information when prediction periods are greater (twelve months). Consequently, they do not perform as well as shallower neural networks

One interesting finding is that while model complexity (number of layers) increases, the economic value of neural networks (annualized Sharpe ratio of the spread portfolio) does not grow monotonically. For instance, NN3 performed best throughout the twelve-month forecast period, while the more complicated NN7 and NN8 fared worse; in the one-month forecast period, NN7 performed best, but NN8 did not further enhance the forecast performance.

While our analysis primarily relies on a measure of economic value derived from machine learning-based portfolios, we acknowledge that once one moves beyond purely statistical criteria for evaluating forecast accuracy, numerous approaches exist for defining and assessing economic value (Leitch and Tanner, 1991). In this regard, we do not assert that our study provides a definitive answer to the broader economic question of whether macroeconomic fundamentals can systematically predict exchange rates. Rather, we argue that employing alternative evaluation metrics based on machine learning portfolios offers a different perspective on the relationship between exchange

rates and fundamentals. This approach may highlight aspects of this relationship—or the absence thereof—that conventional statistical measures fail to capture. Another way to interpret our study is as an exploratory exercise in financial applications, though we recognize certain limitations that should be kept in mind. Notably, we do not incorporate transaction costs, such as bid-ask spreads, into our analysis. That said, the core strategy examined in this paper—a simple zero-net-investment approach requiring only two transactions for each period, one at the beginning and one at the end of each month—is unlikely to be significantly impacted by such costs (Abhyankar et al., 2005). The efficient conversion of prediction capabilities into profitable trading strategies is still a major problem, despite the fact that neural network techniques can offer high prediction accuracy and considerable economic value. Actual returns will be impacted by a number of factors, including capital constraints, transaction costs, and market liquidity. Therefore, even if neural networks are clearly superior in statistical prediction, more study is still needed to determine how to best integrate them into trading methods that are both reliable and rewarding.

Table 5: **Monthly Out-of-Sample Prediction Performance (Mean Return, Standard Deviation, and Sharpe Ratio)**

This table presents the monthly prediction performance at the currency level, evaluating the mean return, standard deviation, and annualiazed Sharpe ratio for different models. The data is divided into training, validation, and testing samples. The model is trained and validated on the training and validation data, respectively, and then applied to the test data without further adjustment. The evaluation includes OLS-based models, neural networks (NNs), dimension reduction models, and tree-based models. A higher Sharpe ratio indicates better risk-adjusted performance.

Prediction Horizon: one month

Portfolio		OLS			OLSH			NN1			NN2			NN3	
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
p1	0.002	0.027	0.271	0.003	0.028	0.346	0.001	0.044	0.050	-0.007	0.041	-0.575	-0.010	0.044	-0.765
p2	-0.001	0.032	-0.073	-0.019	0.056	-1.184	-0.001	0.032	-0.154	-0.016	0.053	-1.022	0.001	0.028	0.068
p3	-0.025	0.063	-1.379	-0.004	0.043	-0.285	-0.009	0.042	-0.754	-0.005	0.043	-0.384	-0.004	0.044	-0.331
p4	-0.007	0.033	-0.726	-0.005	0.035	-0.459	0.000	0.021	0.073	-0.001	0.025	-0.101	0.001	0.030	0.090
p5	0.007	0.025	0.981	0.003	0.022	0.518	-0.005	0.043	-0.406	0.003	0.023	0.383	-0.005	0.039	-0.415
hml	0.005	0.025	0.682	0.001	0.026	0.073	-0.006	0.057	-0.344	0.009	0.037	0.885	0.005	0.056	0.301
Portfolio		NN4			NN5			NN6			NN7			NN8	
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
p1	0.008	0.026	1.083	-0.008	0.044	-0.640	-0.012	0.051	-0.792	-0.007	0.040	-0.629	0.006	0.029	0.695
p2	0.000	0.034	0.003	-0.002	0.036	-0.217	-0.003	0.038	-0.275	-0.011	0.044	-0.835	-0.000	0.021	-0.043
p3	-0.012	0.046	-0.922	-0.005	0.036	-0.482	-0.005	0.054	-0.337	0.005	0.025	0.746	-0.044	0.053	-2.828
p4	0.001	0.018	0.162	-0.006	0.046	-0.445	-0.006	0.024	-0.837	-0.014	0.056	-0.894	0.008	0.026	1.083
p5	-0.014	0.056	-0.885	-0.002	0.038	-0.190	0.002	0.022	0.317	0.006	0.022	0.905	0.009	0.021	1.551
hml	-0.023	0.052	-1.518	0.006	0.051	0.415	0.014	0.049	0.958	0.013	0.041	1.103	0.003	0.019	0.644
Portfolio		PLS			PCR			RF			GBRT				
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR			
p1	0.004	0.024	0.619	0.004	0.025	0.608	0.004	0.024	0.537	0.002	0.028	0.224			
p2	-0.023	0.055	-1.453	-0.025	0.059	-1.502	-0.020	0.059	-1.172	0.007	0.026	0.988			
p3	-0.002	0.042	-0.183	-0.003	0.039	-0.286	-0.004	0.032	-0.418	-0.023	0.056	-1.416			
p4	0.001	0.020	0.210	-0.001	0.034	-0.061	0.004	0.028	0.440	-0.001	0.031	-0.114			
p5	0.002	0.036	0.151	0.001	0.029	0.122	-0.007	0.045	-0.501	-0.010	0.047	-0.772			
hml	-0.003	0.034	-0.278	-0.003	0.030	-0.386	-0.010	0.043	-0.836	-0.012	0.044	-0.975			

Prediction Horizon: twelve month

Portfolio		OLS			OLSH			NN1			NN2			NN3	
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
p1	-0.020	0.050	-1.422	0.003	0.028	0.411	-0.020	0.055	-1.294	-0.009	0.048	-0.684	-0.020	0.055	-1.243
p2	-0.001	0.026	-0.155	-0.012	0.045	-0.931	-0.000	0.026	-0.028	0.005	0.022	0.818	-0.004	0.033	-0.463
p3	-0.013	0.052	-0.850	-0.016	0.049	-1.103	-0.010	0.042	-0.807	-0.002	0.026	-0.276	-0.017	0.052	-1.112
p4	-0.004	0.031	-0.445	-0.002	0.024	-0.355	-0.012	0.042	-0.945	-0.002	0.022	-0.378	0.000	0.021	0.017
p5	0.000	0.028	0.008	-0.014	0.048	-1.011	0.003	0.027	0.357	-0.004	0.038	-0.389	0.003	0.021	0.448
hml	0.021	0.055	1.297	-0.017	0.047	-1.270	0.023	0.058	1.382	0.005	0.056	0.323	0.023	0.053	1.476
Portfolio		NN4			NN5			NN6			NN7			NN8	
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR
p1	0.002	0.027	0.276	0.003	0.027	0.439	-0.020	0.059	-1.196	0.002	0.025	0.245	-0.000	0.021	-0.007
p2	-0.001	0.029	-0.064	-0.019	0.053	-1.203	-0.000	0.031	-0.015	0.002	0.027	0.308	-0.003	0.039	-0.306
p3	-0.016	0.048	-1.200	-0.002	0.026	-0.327	-0.013	0.045	-1.026	-0.036	0.063	-2.007	-0.036	0.063	-1.986
p4	-0.001	0.021	-0.226	0.002	0.020	0.424	-0.001	0.024	-0.089	-0.002	0.026	-0.299	0.003	0.020	0.573
p5	-0.024	0.057	-1.454	-0.005	0.033	-0.513	-0.002	0.030	-0.187	-0.004	0.035	-0.434	-0.003	0.028	-0.361
hml	-0.026	0.055	-1.646	-0.008	0.034	-0.852	0.019	0.059	1.088	-0.006	0.032	-0.658	-0.003	0.026	-0.384
Portfolio		PLS			PCR			RF			GBRT				
	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR	Mean	SD	SR			
p1	0.006	0.022	0.987	-0.003	0.034	-0.327	-0.003	0.031	-0.370	0.002	0.022	0.242			
p2	-0.027	0.058	-1.620	-0.026	0.060	-1.521	-0.029	0.059	-1.738	-0.007	0.041	-0.631			
p3	-0.014	0.048	-1.021	-0.006	0.037	-0.589	-0.004	0.033	-0.406	-0.021	0.055	-1.350			
p4	-0.004	0.034	-0.367	-0.004	0.036	-0.376	-0.000	0.026	-0.009	-0.002	0.036	-0.224			
p5	0.000	0.023	0.030	0.004	0.022	0.581	0.002	0.028	0.213	0.000	0.023	0.021			
hml	-0.006	0.021	-1.003	0.007	0.031	0.758	0.005	0.033	0.531	-0.001	0.020	-0.236			

Our analysis highlights the inherent challenges in achieving robust out-of-sample predictive performance, particularly as the forecasting horizon extends. A key observation from our results is the presence of overfitting in models that exhibit strong in-sample performance, underscoring the critical importance of implementing strategies that enhance generalization. Overall, the results present a nuanced picture. From a statistical standpoint, the performance of different models varies depending on the specific evaluation metric considered. When assessing predictive accuracy through linear correlation between forecasts and actual returns, the best-performing models are OLSH and NN2. However, when shifting the focus to explanatory power in terms of capturing variance in returns, NN7 demonstrates the strongest performance. From an economic perspective, our analysis of portfolio-based investment strategies reveals that NN6 and NN8 yield the highest annualized Sharpe ratios for one-month prediction when applied to the spread machine learning portfolio. This divergence across evaluation criteria is not unexpected, as different models are trained with distinct objective functions that prioritize different aspects of prediction quality. Nevertheless, despite these variations in statistical and economic performance, a consistent theme emerges—neural network-based models tend to exhibit superior predictive ability relative to OLS, suggesting their potential advantage in capturing complex return dynamics.

The maximum drawdown of each model is generally modest within a short prediction horizon (one month), and the drawdown grows overall when the prediction horizon is increased, as Table 6 demonstrates. Furthermore, based on the performance of various models, Neural Networks has a much lower maximum drawdown than other approaches. This could be attributable to either the model's strong fit to market data or its higher risk tolerance under extreme market fluctuations, which would result in less losses in extreme circumstances.

We also looked at the cumulative returns of the top quintile (P5) and bottom quintile (P1) portfolios, which represent the strategy's long and short sides, respectively, when visualizing these market-neutral portfolios. First, we find that models like NN1, NN8, PLS, and NNs have notable yield performance benefits in the long (P5) portfolio. Nonetheless, the returns of these long-term portfolios have exhibited a very flat trend over the last five years, which is in line with the findings of earlier research (Gu et al., 2020). This might be an indication of shifting market dynamics, such a decline in risk premia or a deterioration in the model's capacity to adjust to more recent data. Furthermore, several tree-based models have lately undergone severe retracement (sharply

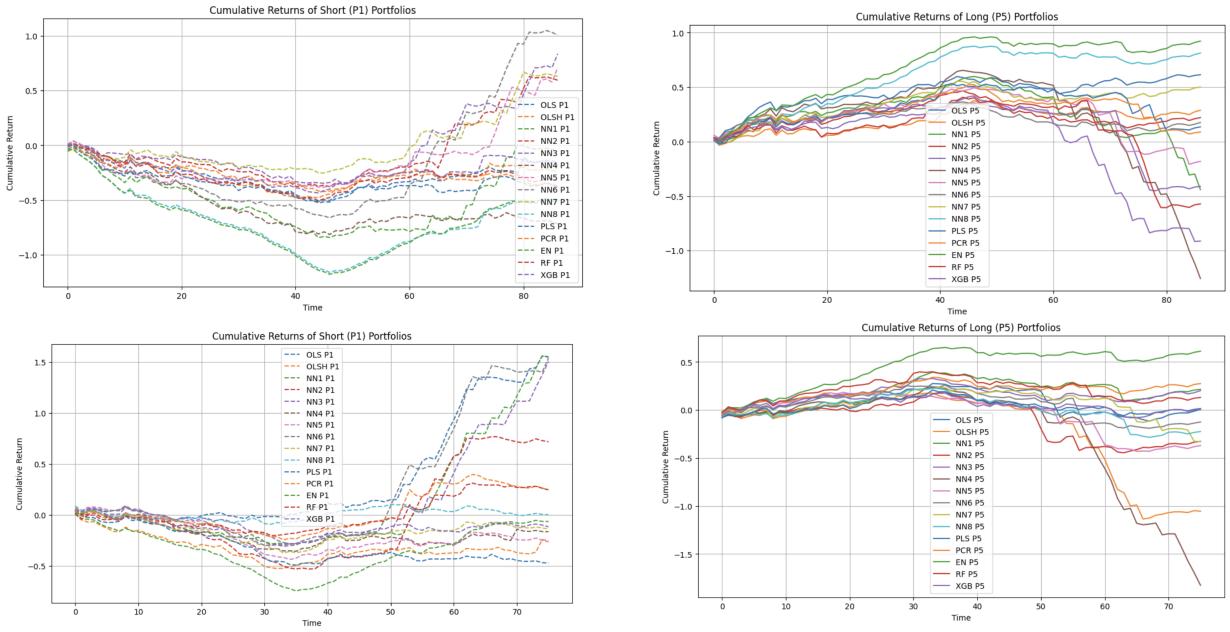
Table 6: **Drawdown comparison across different prediction horizons**
This table compares the drawdowns of equally weighted market-neutral portfolios constructed from different models under two prediction horizons. The left panel presents results for the shorter horizon of one month, while the right panel shows results for the longer horizon. Drawdown is defined as: Drawdown $= \max_{t_1<t_2}(r_{t_1} - r_{t_2})$, where r represents the cumulative log return.

Model	Drawdown (One month)	Drawdown (Twelve months)
OLS	0.154	0.141
OLSH	0.291	1.417
NN1	0.598	0.143
NN2	0.107	0.620
NN3	0.764	0.210
NN4	2.030	2.025
NN5	0.436	0.807
NN6	0.437	0.254
NN7	0.089	0.481
NN8	0.278	0.417
PLS	0.379	0.431
PCR	0.425	0.112
RF	1.008	0.130
GBRT	1.122	0.285

entering negative territory) , suggesting that these approaches are comparatively unstable under the market dynamics of the more recent era. However, the model performance disparity is more noticeable in the short (P1) combination. Certain neural network (NN) models are showing a slow flattening of their cumulative returns. Other models, particularly tree-based models, have, however, substantially returned to the positive range, suggesting that their predictions for the most recent market may have structural biases that cause the short side's performance to diverge from the anticipated direction.

Figure 3: Cumulative Returns of Portfolios Based on the Prediction Model

Based on a predictive model, this graph displays the cumulative log return of a portfolio's out-of-sample forecast performance. One-month forecast periods are represented by the top two curves, and twelve-month prediction periods by the lower two curves. P1 (the poorest quintile) is on the left, while P5 (the top quintile) is on the right. Every portfolio has the same weight.



7. Interpretability analysis

Possible challenges arise despite the numerous advantages of machine learning. While it allows for the inclusion of covariates, offers flexibility in functional forms, accommodates nonlinearity, and leverages multidimensional advantages, there are notable bottlenecks in its application. One such bottleneck pertains to the interpretability of machine learning models, often referred to as the "black-box" problem. However, this challenge is not insurmountable. Techniques such as permutation importance, as highlighted by Gu et al. (2020), can reveal which covariates are most crucial in explaining expected returns. Furthermore, various local and global interpretability approaches can be employed to address this issue, ensuring that the insights gleaned from machine learning models are both understandable and actionable.

The black-box characteristic of machine learning, which refers to the impossibility to map parameters to specific output characteristics, poses an issue for its reliability even in the case of sped up calibration satisfactory accuracy levels. The term 'interpretability' lacks a precise definition, but Miller (2019) offers a non-mathematical definition: *'the degree to which a human can understand the cause of a decision'*. The more interpretable a machine learning model, the simpler to understand why specific judgements or predictions have been made. When building a model, not only is its accuracy fundamental, but also how these outputs are derived from inputs and how stable this mapping is. Thus, interpretability can be advantageous in two cases: (1) with a thorough understanding of the model, one may test whether the map from inputs to outputs corresponds to intuitive understanding; (2) with a lack of model expertise, one may employ interpretability models to increase the comprehension of the model. Overall, two machine learning interpretability classifications have been devised by Molnar (2020): local and global.

7.1. Local interpretability

Local interpretability models aim to explain how a machine learning model generate its prediction for a single instance (Molnar, 2020), rather than providing a global understanding of the model's overall behavior. This approach is particularly valuable when working with complex models like neural networks, where the relationship between inputs and predictions can be opaque. The core idea behind local interpretability is to simplify the input and examine how individual components contribute to the final prediction. This method allows us to answer practical questions such

as, "What is the role of each characteristic in determining this specific prediction?" or "How would the prediction change if a certain characteristic were removed?"

Now, let us consider how local interpretability is formally defined (Yuan et al., 2024). A local interpretability model, often denoted as g, approximates the behavior of the machine learning model $\widehat{f}$ in the vicinity of a specific input x. For a simplified input x', the interpretability model g aims to replicate the predictions of $\widehat{f}$ for inputs that are close to x'. This can be expressed informally as $g(z') \approx \widehat{f}(h_x(z'))$ for simplified inputs z' that are similar to x'. To ensure the approximation is meaningful, a more precise version of this relationship might involve defining neighborhoods around x' and setting thresholds for how close the predictions need to be.

This is a characteristic attribution framework for intrepretability. In this framework, the prediction is decomposed into a baseline value, ϕ_0, which represents the prediction when all characteristics are inactive, and a sum of contributions from each characteristic, denoted as ϕ_i. Mathematically, this can be expressed as:

$$g(z') = \phi_0 + \sum_i z'_i \phi_i,$$

where z'_i represents if characteristic i is active (1) or inactive (0). For example, if only the first and third characteristics are active ($z' = [1, 0, 1]$), the prediction is determined by the baseline value plus the contributions of these two characteristics. This additive structure makes it easy to see how much each characteristic contributes to the overall prediction.

7.2. Global interpretability

Instead of focusing on a single example, global interpretability examines the overall impact of characteristics throughout the sample. An overview of the feature contribution is given by the global approach.

Shapley value in game theory is introduced by global interpretability, which assigns each player the gain or loss attribute of a multiplayer game in order to gauge how each player affects the outcome of the game. In our context, participants are characteristics, the game is the prediction task, and the gain is the prediction. The Shapley value provides a thorough assessment of feature relevance by quantifying the contribution of each characteristic while taking into account all potential subsets.

To formalize this concept (Yuan et al., 2024), consider a scenario with n players, represented as a set $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$. For any subset $G \subseteq \mathcal{P}$, the function $f(G)$ measures the collective

contribution of the players in G. The Shapley value for a specific player $p_k \in \mathcal{P}$ is defined as:

$$\phi_k = \sum_{G \subseteq \mathcal{P} \setminus \{p_k\}} \frac{|G|!\,(|\mathcal{P}| - |G| - 1)!}{|\mathcal{P}|!} \Big(f(G \cup \{p_k\}) - f(G) \Big),$$

where $|G|$ denotes the size of the coalition G, and $|\mathcal{P}| = n$ is the total number of players. This formula systematically evaluates the marginal contribution of player p_k across all possible subsets G that exclude p_k, averaging these contributions to provide a fair attribution. Intuitively, this approach considers every possible coalition the player might join and calculates their added value in each scenario.

In machine learning, this concept translates seamlessly to characteristic attribution. Each input characteristic is treated as a "player," and the model's prediction for a specific instance is the outcome of the "game." The Shapley value for a characteristic represents its average contribution to the prediction, taking into account all possible combinations of active and inactive characteristics. For a dataset with M characteristics, the Shapley value for characteristic k is given by:

$$\phi_k \left(\widehat{f}, x \right) = \sum_{We \subseteq \{1,2,\ldots,M\} \setminus \{k\}} \frac{|I|!\,(M - |I| - 1)!}{M!} \Big(\widehat{f}\big(h_x(z'_{We} \cup \{k\})\big) - \widehat{f}\big(h_x(z'_I)\big) \Big),$$

where z'_I is a binary vector indicating which characteristics are active in the subset I, and h_x maps this simplified representation back to the original input space. For example, if z'_I indicates that only the first and third characteristics are active, $h_x(z'_I)$ reconstructs an input vector where all other characteristics are replaced with their average values. This ensures that the Shapley value captures the marginal effect of each characteristic in a systematic and unbiased manner.

Calculating Shapley values in practice can be computationally difficult because it requires considering all possible subsets of characteristics, which grows exponentially with the number of features. For a dataset with M characteristics, there are 2^M subsets, making exact computation infeasible for high-dimensional data. To address this issue, various approximation algorithms have been developed. One of the most commonly used tools is `SHAP` ((Shapley Additive exPlanations, package in Python)). This approach allows to rank characteristics based on their average importance, providing insights into which characteristics have the greatest overall impact on the model's predictions.

There are drawbacks to the Shapley value technique as well. Computational flexibility is one

issue that was previously mentioned. Its use to large-scale data is thus limited. We introduce the approximation approach in this way. Another issue is that it makes the unrealistic assumption that characteristics are independent of one another. Real data may be cross-sectionally or serially correlated.

Unlike permutation importance, which evaluates characteristic importance by shuffling characteristic values and measuring the resulting drop in model performance, Shapley values provide a comprehensive attribution by considering all possible subsets of characteristics. Similarly, while partial dependence plots visualize the marginal effect of a single characteristic by averaging predictions over all other characteristics, they may fail to capture interactions between characteristics. Shapley values, by contrast, inherently account for characteristic interactions, making them a more robust choice for models with complex dependencies.

In conclusion, global interpretability offers a systematic method to comprehending the role that distinct features play in machine learning models. Cooperative game theory is introduced in this manner. The widespread use of `SHAP` and related techniques highlights the usefulness of the Shapley value in machine learning, even though computational difficulties and feature independence assumptions still require investigation. Global explainability tools will be crucial in establishing transparency, accountability, and reliability as models grow more intricate especially in some high-stake fields.

7.3. Empirical results of interpretability

The driving factors of returns were analyzed in this study using a range of model interpretation techniques. The DeepLIFT and Layer-wise Relevance Propagation (LRP) approaches were used, respectively, to examine the important factors under one-month and twelve-month forecast periods from the standpoint of local interpretability. The findings demonstrate that, despite the great degree of consistency between the analytical results of the two local interpretability methodologies, there are still distinctions between them and the global interpretability conclusions. First, according to the DeepLIFT method's study results, the value factor is the least significant driving element for the one-month horizon, while the NFA factor is the most significant among the currency-specific variables. Global liquidity, F1 and F2 (primary component variables taken from large-scale macro and financial time series data sets), and TIC1 and TIC2 (inventories) have the least contributions to global features, whereas global liquidity is the most important driving element. This might imply

that, in the short run, shifts in global market conditions and macro liquidity account for a greater portion of exchange rate volatility than do long-term bond market inventories in US Treasuries. The LRP approach's one-month horizon findings are quite similar. In particular, the value component continues to be the least significant currency-specific element, whilst long-term momentum and NFA are the most significant. PTIC1 and PTIC2 have the least influence on global features, while F1 and F2 (principal component of macro) are the most significant driving variables. While long-term bond market-related factors have a very little impact on short-term estimates, this result further solidifies the pivotal role of macro-financial market major component factors on exchange rate fluctuations. The DeepLIFT method for the 12-month horizon indicates that, among currency-specific factors, the carry factor has a leading position, with the value and NFA elements having the least influence. This might be as a result of investors' increased sensitivity to the arbitrage effect of interest rates across various currencies and the longer-lasting effects of carry trading techniques. Furthermore, interest dynamics—which include indicators like libor-tbill, libor-ois, and libor—are the most important component in terms of global features. This suggests that long-term shifts in the interest rate environment are the primary determinants of currency returns. Similar to the DeepLIFT method's conclusion, the LRP method's results under a 12-month horizon indicate that, among currency-specific elements, the momentum component is the most significant.

When compared over several prediction periods, the outcomes of the local interpretability techniques (DeepLIFT and LRP) are very comparable. This result would suggest that the machine learning model's decision-making logic at the single instance level is very stable regardless of the local interpretability approach employed. However, differing degrees of causal chains may be the cause of the distinction between local and global interpretability. The weight ordering of important components may range significantly between local explanations, which base their conclusions more on short-term signals or sample characteristics, and global explanations, which tend to concentrate on long-term trends and the general distribution of data. Moreover, the nonlinear nature of machine learning models could be reflected in this discrepancy. While machine learning models may capture more complicated nonlinear patterns, traditional economic and financial theory often assumes linear correlations between variables. Therefore, local interpretability focuses more on how individual predictions are influenced, while global interpretability emphasizes the overall stability of the model. Our findings demonstrate that, from the standpoint of empirical asset pricing, investors

should concentrate on distinct driver groups at various time periods. While carry trade tactics and global interest rate dynamics are more important elements in the long run, macro liquidity and market principal component characteristics have a greater effect in short-term trading strategies.

From Figures 6 and 7, the characteristics are ranked according to their importance, with those higher up in the charts contributing more significantly to the overall predictive capability of the model. These interaction characteristics dominate the global interpretability of the model. Particularly, characteristics that capture the interactions between individual tradable factors and global factors play a crucial role. The performance of individual tradable factors is often mediated by the broader macroeconomic environment, and the model relies heavily on these interaction characteristics to enhance its predictive capabilities.

For a prediction horizon of one month, the most influential individual factors are carry and short-term momentum, underscoring their dominant role in shorter-term forecasts. These factors likely capture immediate and transient trading dynamics, which are more relevant for short-term predictions. In contrast, for a prediction horizon of twelve months, the key contributors among individual factors shift to carry. This shift suggests that the importance of individual factors evolves with the prediction horizon, as longer-term forecasts may rely more on structural and persistent trends, such as the difference in the interest rate.

From the perspective of global factors, the characteristics with the highest relevance for a one-month horizon are closely tied to liquidity conditions and interest rate dynamics. Examples include the Libor-OIS spread and OIS-TBill spread, which emphasize the importance of liquidity-based indicators derived from Libor, TBill, and OIS data. These characteristics highlight how short-term predictions depend on precise measures of market liquidity and interest rate movements. As the prediction horizon extends to twelve months, the ranking of characteristic importance remains almost the same.

All things considered, the predictors with the greatest explanatory power are the interactions between the global and individual tradable components. Under both short- and long-term predictions, we simultaneously observe changes in the most influential characteristics.

The color gradient in the heatmap, ranging from blue to red, represents the characteristic values from low to high. For the prediction horizon of 1 month, US unemployment rate primarily influences the model through its interactions with short-term momentum and long-term momentum.

Additionally, by analyzing the distribution of SHAP values on either side of zero, we can identify the positive or negative contributions of different characteristics to the model output, which reflects the directionality of characteristic contribution. Most characteristics exhibit SHAP values concentrated predominantly on either the positive or negative side. For instance, for shorter prediction horizons, the topmost four characteristics for the prediction horizon of 1 month, whose SHAP values are overwhelmingly positive. The strong unidirectional distribution of SHAP values for most characteristics indicates that these characteristics have a clear and stable impact on the model, which is unlikely to reverse with changes in input values. In contrast, only a small number of characteristics exhibit a more balanced distribution of SHAP values across both positive and negative sides. This symmetrical distribution may indicate more complex dynamic relationships. For example, these characteristics may contribute positively to the model predictions under certain conditions, while under other conditions, their impact may turn negative. This bidirectional effect suggests that the contribution of such characteristics depends on the specific range of characteristic values or the market environment. These characteristics' dual roles in the model highlight the importance of context-dependent relationships.

Figure 4: Local interpretability for prediction horizon of 1 month

This figure presents the local interpretability results for the best-performing prediction model for a prediction horizon of 1 month, a neural network with seven layers. The DeepLIFT method-based feature contribution ranking, arranged by feature contribution absolute value, is displayed in the top row. Based on the average absolute value of the features, the top 20 and bottom 20 most significant features—including univariate and interaction terms—are listed. This metric shows how much each attribute contributed overall to the model's predicted outcomes. A clear depiction of the significance of the various features in the dataset is provided by the ranking. The LRP method's results are displayed in the second row.

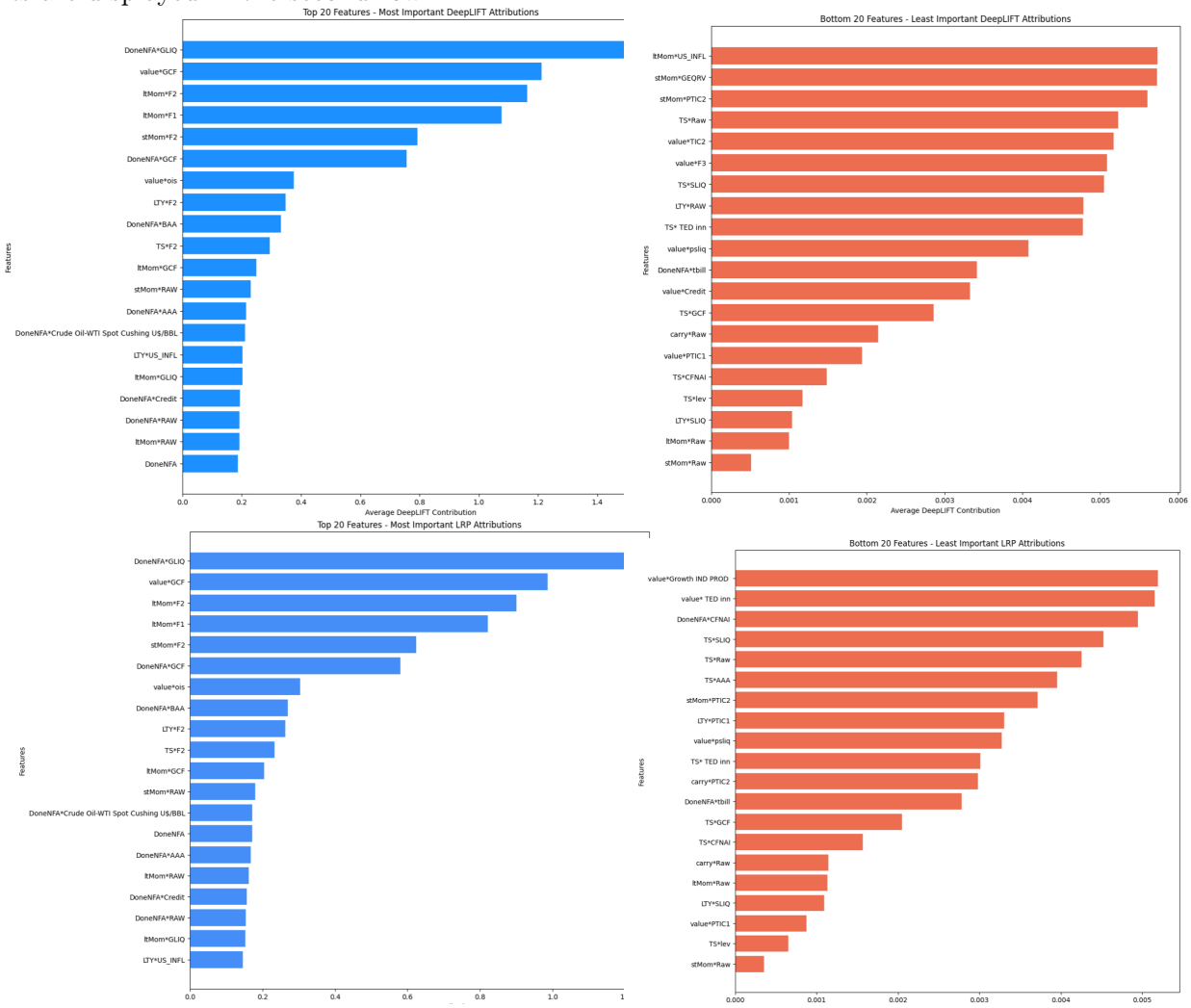


Figure 5: Local interpretability for prediction horizon of 12 months

This figure presents the local interpretability results for the best-performing prediction model for a prediction horizon of 12 months, a neural network with eight layers. The DeepLIFT method-based feature contribution ranking, arranged by feature contribution absolute value, is displayed in the top row. Based on the average absolute value of the features, the top 20 and bottom 20 most significant features—including univariate and interaction terms—are listed. This metric shows how much each attribute contributed overall to the model's predicted outcomes. A clear depiction of the significance of the various features in the dataset is provided by the ranking. The LRP method's results are displayed in the second row.

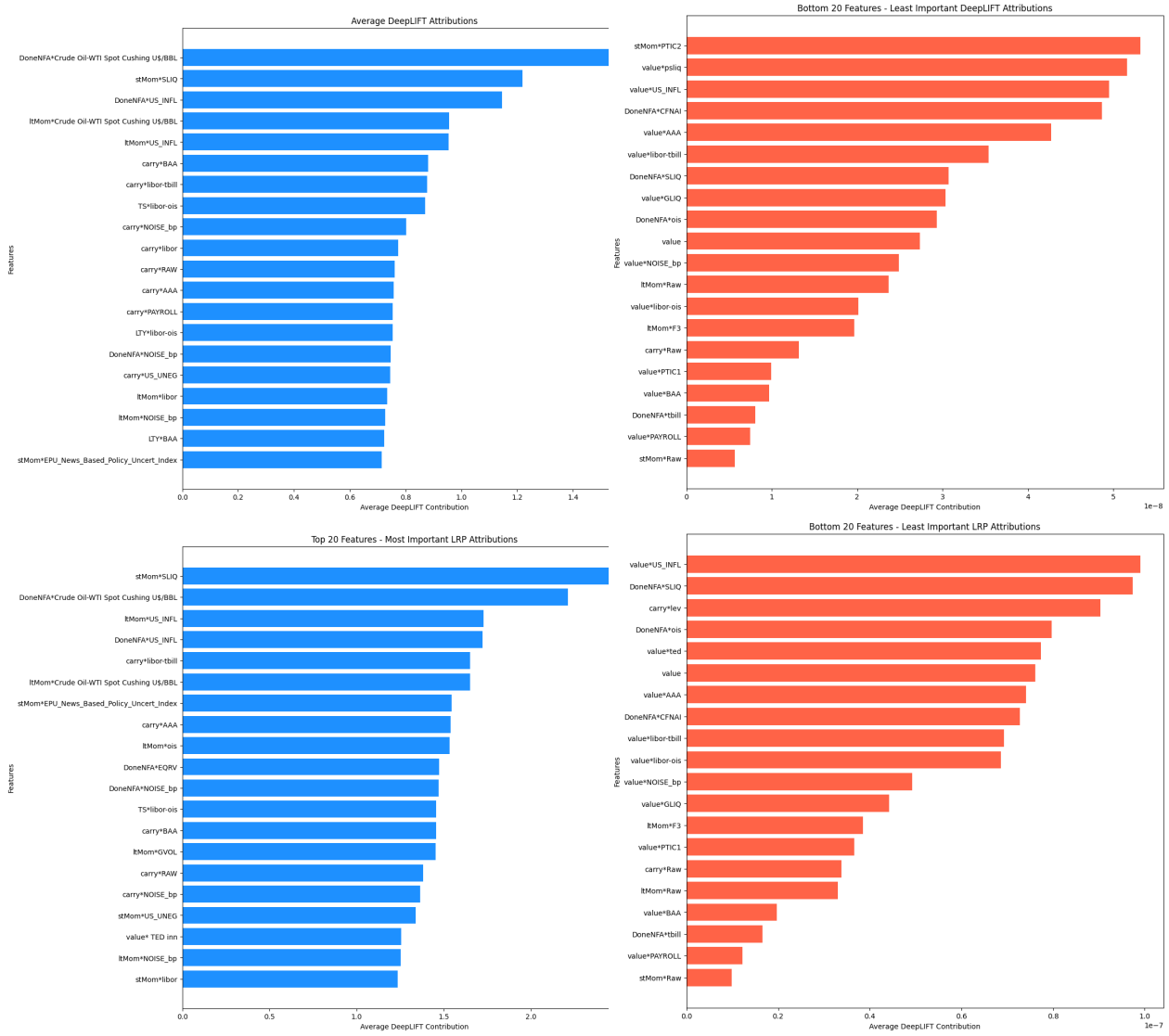


Figure 6: Global interpretability for prediction horizon of 1 month

The global interpretability findings for the best prediction model, a seven-layer neural network, with a one-month forecast horizon are displayed in the figure. Features are ranked on the left side according to their average absolute value. One statistic that measures each feature's contribution to the model's global predictions is the Shapley value. Out of all the features, the top 20 are chosen as the most significant. These attributes encompass both individuals and interactions. A heatmap of the Shapley values for the top 20 characteristics is displayed on the right. Granular analysis is provided via heatmaps, which offer comprehensive insights into the distribution and variability of feature contributions across samples.

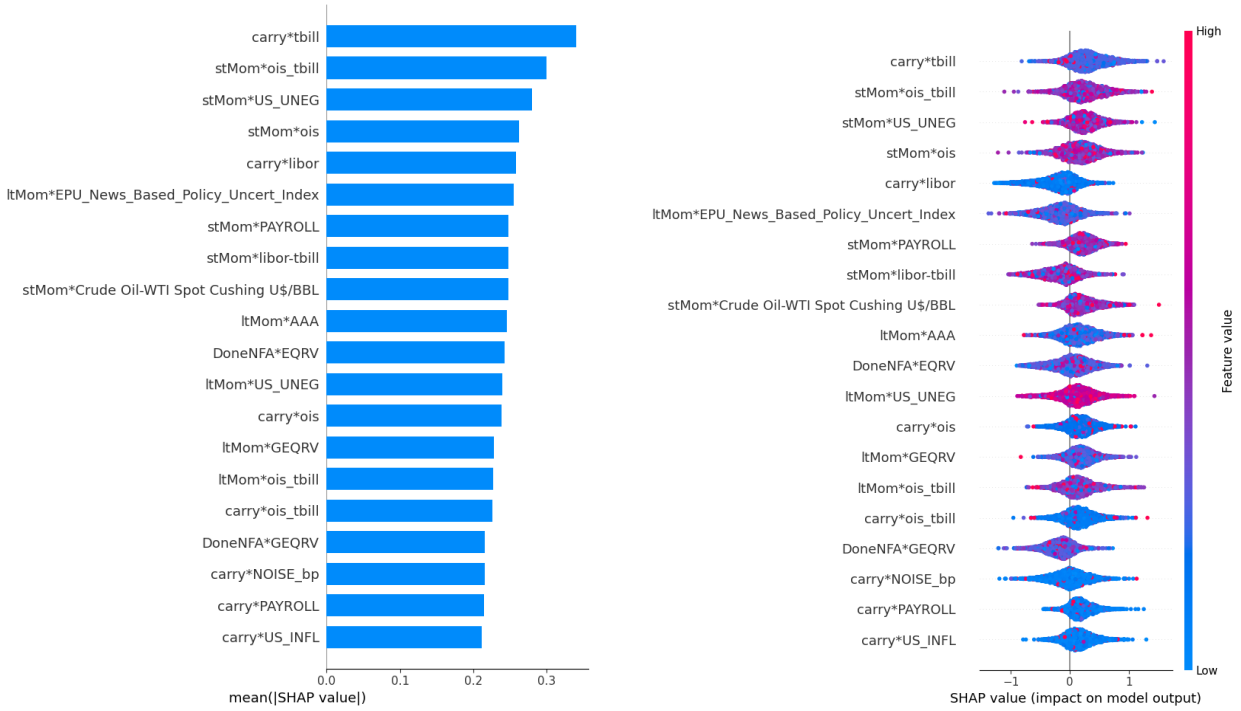
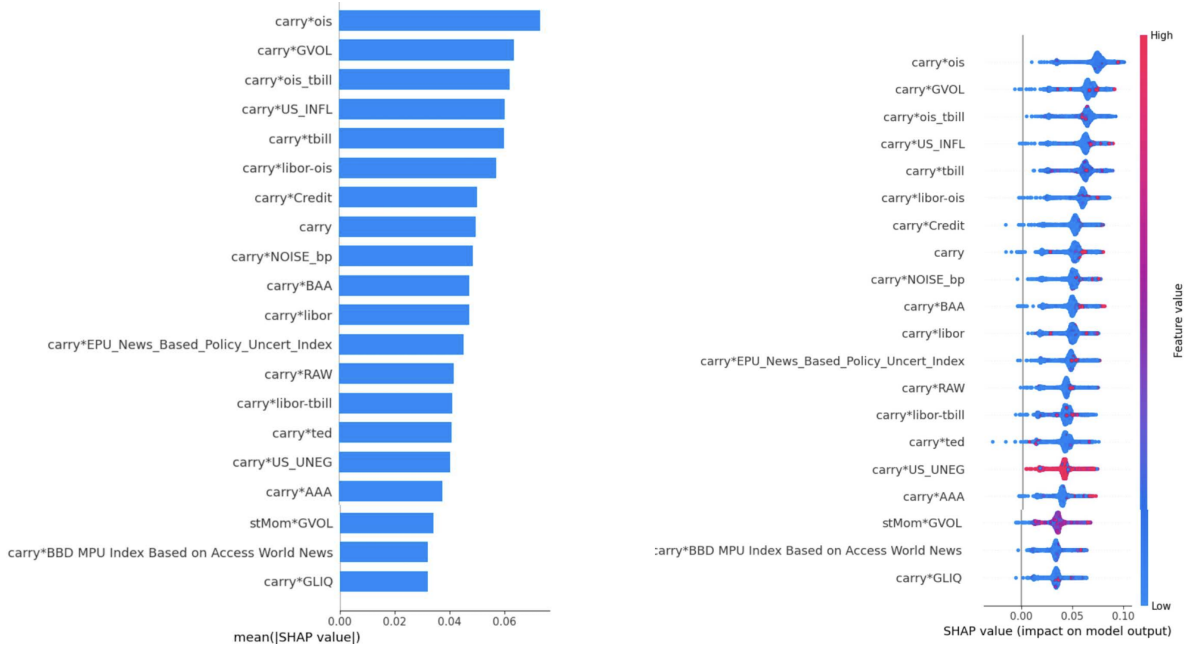


Figure 7: Global interpretability for prediction horizon of 12 months

The global interpretability findings for the best prediction model, an eight-layer neural network, with a twelve-month forecast horizon are displayed in the figure. Features are ranked on the left side according to their average absolute value. One statistic that measures each feature's contribution to the model's global predictions is the Shapley value. Out of all the features, the top 20 are chosen as the most significant. These attributes encompass both individuals and interactions. A heatmap of the Shapley values for the top 20 characteristics is displayed on the right. Granular analysis is provided via heatmaps, which offer comprehensive insights into the distribution and variability of feature contributions across samples.



8. Conclusions

This study represents a pioneering effort in applying machine learning to currency excess return prediction. By allowing for nonlinear predictive functions and accommodating high-dimensional data, our approach incorporates a broader set of information-rich predictor variables than traditional models. A key finding is that neural networks consistently outperform the others across different evaluation metrics and prediction horizons, demonstrating the potential of machine learning in currency return forecasting. A more complex neural network architecture generally enhances its predictive power, which is very similar to the results of the equity study (Gu et al., 2020). Specifically, neural networks are the only models that outperform the random walk benchmark. Economically, they also yield the highest Sharpe ratio when constructing long-short spread portfolios.

To address the well-known black-box issue of machine learning models, we employ DeepLIFT, LRP, and Shapley value analysis to interpret the sources of predictive power. Our results indicate that the dominant predictive signals arise from the interaction between global state variables and currency-specific factors, rather than from any single variable alone. At a global scale, mong tradable factors, carry and momentum emerge as the most powerful predictors, aligning with their well-established roles in currency investment strategies. Among global state predictors, market liquidity stands out as the most influential, highlighting its critical role in shaping return dynamics. But the local interpretbility results are distinct from the global approach. One possible reason could be that the global predictability reflects overall model performance across the entire dataset. These findings enhance our understanding of the risk-return trade-offs inherent in currency investment strategies.

Our results are consistent with prior empirical findings, such as those in Nucera et al. (2024), while making further progress in addressing the challenge of the FX "factor zoo." By systematically identifying the most relevant characteristics for return prediction, we contribute to the effort of refining and streamlining factor-based currency investment strategies. At the same time, the demonstrated success of machine learning algorithms in return prediction offers promising implications for both economic modeling and practical portfolio management. Our findings help justify the growing role of machine learning across the broader fintech industry, supporting its increasing integration into asset pricing, risk management, and investment decision-making processes.

References

Abhyankar, Abhay, Lucio Sarno, and Giorgio Valente, "Exchange rates and fundamentals: evidence on the economic value of predictability," *Journal of International Economics*, 2005, *66* (2), 325–348.

Amat, Christophe, Tomasz Michalski, and Gilles Stoltz, "Fundamentals and exchange rate forecastability with simple machine learning methods," *Journal of International Money and Finance*, 2018, *88*, 1–24.

Andrés, David, Nov 2023.

Asness, Clifford S, Tobias J Moskowitz, and Lasse Heje Pedersen, "Value and momentum everywhere," *The journal of finance*, 2013, *68* (3), 929–985.

Bianchi, Daniele, Matthias Büchner, and Andrea Tamoni, "Bond risk premiums with machine learning," *The Review of Financial Studies*, 2021, *34* (2), 1046–1089.

Breiman, Leo, "Random forests," *Machine learning*, 2001, *45*, 5–32.

_ , *Classification and regression trees*, Routledge, 2017.

Carhart, Mark M, "On persistence in mutual fund performance," *The Journal of finance*, 1997, *52* (1), 57–82.

Chen, Luyang, Markus Pelger, and Jason Zhu, "Deep learning in asset pricing," *Management Science*, 2024, *70* (2), 714–750.

Chernov, Mikhail, Magnus Dahlquist, and Lars Lochstoer, "Pricing currency risks," *The Journal of Finance*, 2023, *78* (2), 693–730.

Della Corte, Pasquale, Steven J Riddiough, and Lucio Sarno, "Currency premia and global imbalances," *The Review of Financial Studies*, 2016, *29* (8), 2161–2193.

Engel, Charles, Nelson C Mark, Kenneth D West, Kenneth Rogoff, and Barbara Rossi, "Exchange rate models are not as bad as you think [with comments and discussion]," *NBER macroeconomics annual*, 2007, *22*, 381–473.

Fama, Eugene F, "Forward and spot exchange rates," *Journal of monetary economics*, 1984, *14* (3), 319–338.

_ and Kenneth R French, "Common risk factors in the returns on stocks and bonds," *Journal of financial economics*, 1993, *33* (1), 3–56.

Filippou, Ilias, David Rapach, Mark P Taylor, and Guofu Zhou, "Exchange rate prediction with machine learning and a smart carry trade portfolio," 2020.

_ , _ , _ , and _ , "Economic Fundamentals and Short-Run Exchange Rate Prediction: A Machine Learning Perspective," *Available at SSRN 3455713*, 2023.

Friedman, Jerome H, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, 2001, pp. 1189–1232.

Giglio, Stefano, Bryan Kelly, and Dacheng Xiu, "Factor models, machine learning, and asset pricing," *Annual Review of Financial Economics*, 2022, *14* (1), 337–368.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, 2020, *33* (5), 2223–2273.

Hansen, Lars Peter and Robert J Hodrick, "Forward exchange rates as optimal predictors of future spot rates: An econometric analysis," *Journal of political economy*, 1980, *88* (5), 829–853.

Hassan, Tarek, Thomas M Mertens, and Jingye Wang, "A currency premium puzzle," Technical Report 2024.

Hastie, Trevor, "The elements of statistical learning: data mining, inference, and prediction," 2009.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani et al., *An introduction to statistical learning*, Vol. 112, Springer, 2013.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, "Characteristics are covariances: A unified model of risk and return," *Journal of Financial Economics*, 2019, *134* (3), 501–524.

Leippold, Markus, Qian Wang, and Wenyu Zhou, "Machine learning in the Chinese stock market," *Journal of Financial Economics*, 2022, *145* (2), 64–82.

Leitch, Gordon and J Ernest Tanner, "Economic forecast evaluation: profits versus the conventional error measures," *The American Economic Review*, 1991, pp. 580–590.

Lintner, John, "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets," *The Review of Economics and Statistics*, 1965, pp. 13–37.

Lundberg, Scott M and Su-In Lee, "A Unified Approach to Interpreting Model Predictions, Nov," *arXiv preprint arXiv:1705.07874*, 2017.

Lustig, Hanno and Adrien Verdelhan, "The cross section of foreign currency risk premia and consumption growth risk," *American Economic Review*, 2007, *97* (1), 89–117.

_ , Nikolai Roussanov, and Adrien Verdelhan, "Common risk factors in currency markets," *The Review of Financial Studies*, 2011, *24* (11), 3731–3777.

Mark, Nelson C, "Exchange rates and fundamentals: Evidence on long-horizon predictability," *The American Economic Review*, 1995, pp. 201–218.

Massy, William F, "Principal components regression in exploratory statistical research," *Journal of the American Statistical Association*, 1965, *60* (309), 234–256.

Meese, Richard A and Kenneth Rogoff, "Empirical exchange rate models of the seventies: Do they fit out of sample?," *Journal of international economics*, 1983, *14* (1-2), 3–24.

Menkhoff, Lukas, Lucio Sarno, Maik Schmeling, and Andreas Schrimpf, "Carry trades and global foreign exchange volatility," *The Journal of Finance*, 2012, *67* (2), 681–718.

_ , _ , _ , and _ , "Currency momentum strategies," *Journal of Financial Economics*, 2012, *106* (3), 660–684.

_ , _ , _ , and _ , "Currency value," *The Review of Financial Studies*, 2017, *30* (2), 416–441.

Miller, Tim, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, 2019, *267*, 1–38.

Molnar, Christoph, *Interpretable machine learning*, Lulu. com, 2020.

Nagel, Stefan, *Machine learning in asset pricing*, Vol. 1, Princeton University Press, 2021.

Newey, Whitney K and Kenneth D West, "Hypothesis testing with efficient method of moments estimation," *International Economic Review*, 1987, pp. 777–787.

Nucera, Federico, Lucio Sarno, and Gabriele Zinna, "Currency risk premiums redux," *The Review of Financial Studies*, 2024, *37* (2), 356–408.

Pakkanen, Mikko, "Lecture Notes on Deep Learning," December 2021. Unpublished course materials, Imperial college London.

Pfahler, Jonathan Felix, "Exchange rate forecasting with advanced machine learning methods," *Journal of Risk and Financial Management*, 2021, *15* (1), 2.

Ross, Stephen A, "The arbitrage theory of capital asset pricing," *Journal of Economic Theory*, 1976, *13* (3), 341–360.

Sharpe, William F, "Capital asset prices: A theory of market equilibrium under conditions of risk," *The journal of finance*, 1964, *19* (3), 425–442.

Wada, Tatsuma, "Out-of-sample forecasting of foreign exchange rates: The band spectral regression and LASSO," *Journal of International Money and Finance*, 2022, *128*, 102719.

Wold, H, "Estimation of principal components and related models by iterative least squares," 1966.

Yaohao, Peng and Pedro Henrique Melo Albuquerque, "Non-linear interactions and exchange rate prediction: Empirical evidence using support vector regression," *Applied Mathematical Finance*, 2019, *26* (1), 69–100.

Yuan, Bo, Damiano Brigo, Antoine Jacquier, and Nicola Pede, "Deep learning interpretability for rough volatility," *arXiv preprint arXiv:2411.19317*, 2024.

Appendix A. Global characteristic

Factor	Description
	Financial Factors
MOVE	Volatility index measuring options on U.S. Treasury bonds, reflecting uncertainty in fixed-income markets.
VXO	A measure of implied volatility in the S&P 100, often used as a proxy for market risk perception.
MF1, MF2, MF3	Principal components derived from a broad dataset of economic and financial indicators.
GVOL, GLIQ	Global foreign exchange market volatility and liquidity indices constructed from daily trading data.
PSLIQ	Liquidity measure for equities, assessing market-wide ease of trading conditions.
SLIQ	A systematic indicator of liquidity variations in the FX market, emphasizing low-frequency trends.
TED	The spread between interbank lending rates and Treasury bills, serving as a gauge of credit risk.
NOISE	An indicator of arbitrage capital availability based on deviations in U.S. Treasury bond prices.
ICAP	Captures fluctuations in financial intermediaries' equity capital, impacting asset pricing dynamics.
OILVOL	Volatility of crude oil prices, estimated via historical fluctuations in daily returns.
GCF	A synthetic factor representing global financial conditions, constructed using price movements of various risky assets.
TIC	Official and private inventory levels in U.S. Treasuries, standardizing across time for consistency.
CORP	The yield differential between investment-grade and lower-rated corporate bonds, signaling credit spreads.
LIB-OIS	A measure of stress in the interbank market derived from differences in short-term borrowing rates.

EQRV	Realized volatility in the S&P 500 index, computed using daily price movements.
GEQRV	A factor capturing worldwide equity market volatility based on stock market indices.
	Macro Factors
IPUS	Monthly industrial production growth rate, indicating economic activity in the U.S.
CPIUS	Consumer price inflation rate in the U.S., reflecting changes in cost-of-living metrics.
NFPYR	Change in non-farm payroll employment, a widely followed labor market indicator.
CFNAWe	A composite index summarizing national economic conditions using multiple macro indicators.
UNEUS	Unemployment rate changes in the U.S., serving as a key labor market barometer.
CUS	Household consumption expenditure trends, capturing shifts in consumer behavior.
IPW	A global measure of industrial production growth, aggregated across multiple economies.
CPIW	A composite inflation metric, combining price index data from different countries.
UNEW	A weighted measure of unemployment trends across major economies.
IPSTD	Cross-country dispersion of industrial production changes, reflecting economic heterogeneity.
CPISTD	A measure of inflation variability across countries, highlighting disparities in price stability.
	Text-Based Factors
GEPU	A macroeconomic policy uncertainty index aggregating country-specific data using weighted GDP.

FSWe	A financial stress indicator computed from media coverage and market conditions.
EMV	A textual-based equity market volatility measure derived from news sources.
EPU	An index measuring economic policy uncertainty by analyzing newspaper articles and policy discussions.

Appendix B. Details: Neural Networks (NN)

The training Procedure can be summarised as:

- Initialization: The weights $\mathbf{W}_k$ are typically initialized randomly, biases $\mathbf{b}_k$ are usually initialized to zero or small random values.
- Forward Pass: Given the input $\mathbf{h}_0$, the model computes the layer-wise results sequentially using the formulas above, ultimately producing $\hat{y}$.
- Loss Computation: The difference between the predicted output $\hat{y}$ and the actual target y is quantified using a loss function, such as mean squared error (for regression) or cross-entropy (for classification).
- Backward Pass: Using backpropagation, the gradients of the loss with respect to the parameters $\mathbf{W}_k$ and $\mathbf{b}_k$ are computed.
- Optimization: The parameters are updated using an optimization algorithm, typically by making incremental moves along the negative gradient, as determined by the learning rate.
- Iteration: The process repeats for a set number of iterations or until convergence is achieved, such as when the loss function stops improving significantly.

This detailed process underscores the interplay between parameters, hyperparameters, and the role of non-linear transformations in enabling neural networks to learn and generalize from data effectively.

Appendix C. Details: Principal Components regression (PCR) and Partial least squares (PLS)

The procedure of PCR can be summarised as below.

- PCA on predictors:

  Given a dataset with NT observations and p predictor variables, we define the centered and standardised predictor matrix:

$$X_{\text{std}} \in \mathbb{R}^{NT \times p}, \quad \text{where } \sum_{i=1}^{NT} (X_{\text{std}})_{ij} = 0, \quad \forall j \in \{1, \ldots, p\}.$$

  The sample covariance matrix of X_{std} is:

$$\Sigma_{X_{\text{std}}} = \frac{1}{NT-1} X_{\text{std}}^T X_{\text{std}} \in \mathbb{R}^{p \times p}.$$

  The principal components are found by solving the eigenvalue problem:

$$\Sigma_{X_{\text{std}}} v_j = \lambda_j v_j, \quad j = 1, \ldots, p,$$

  where v_j are the eigenvectors (principal directions) and λ_j are the corresponding eigenvalues. The eigenvectors are stacked in a matrix:

$$V = [v_1, v_2, \ldots, v_p] \in \mathbb{R}^{p \times p}.$$

  The principal component transformation is then given by:

$$Z = X_{\text{std}} V \in \mathbb{R}^{NT \times p},$$

  where the columns of Z are the principal components (PCs), sorted in descending order of variance.

  To reduce dimensionality, we retain only the first k principal components:

$$Z_k = X_{\text{std}} V_k \in \mathbb{R}^{NT \times k},$$

where $V_k \in \mathbb{R}^{p\times k}$ contains only the first k principal directions.

- Regression on principal components:

  Instead of regressing r directly on X_{std}, we perform regression on Z_k:

  $$r = Z_k\beta + \varepsilon,$$

  where $\beta \in \mathbb{R}^k$ are the regression coefficients and ε is the error term. The least squares estimate of β is:

  $$\hat{\beta} = (Z_k^T Z_k)^{-1} Z_k^T r.$$

  The final prediction is obtained as:

  $$\hat{r} = Z_k\hat{\beta} = X_{\text{std}}V_k(V_k^T X_{\text{std}}^T X_{\text{std}} V_k)^{-1} V_k^T X_{\text{std}}^T r.$$

The procedure of PLS can be summarised as below.

- Finding latent components:

  PLS finds components $S \in \mathbb{R}^{NT\times k}$ such that they maximize covariance of X_{std} and r. The decomposition of X_{std} and r is given by:

  $$X_{\text{std}} = SP^T + E_{\text{pls}}, \quad r = Sq + f_{\text{pls}},$$

  where $S \in \mathbb{R}^{NT\times k}$ is the matrix of latent scores, $P \in \mathbb{R}^{p\times k}$ and $q \in \mathbb{R}^k$ are loadings, $E_{\text{pls}} \in \mathbb{R}^{NT\times p}$ and $f_{\text{pls}} \in \mathbb{R}^{NT}$ are residuals.

- Maximizing Covariance:

  The latent components are computed by solving:

  $$\max_S \operatorname{cov}(S, r) = \max_S \|S^T r\|^2,$$

  subject to $S = X_{\text{std}}V$, where V contains weights that define the projection.

PLS finds the first component s_1 by maximizing:

$$v_1 = \arg\max_v \operatorname{cov}^2(X_{\text{std}}v, r), \quad \text{subject to } \|v\| = 1.$$

This is solved iteratively by Singular Value Decomposition (SVD):

$$U, D, V_{\text{svd}} = \text{SVD}(X_{\text{std}}^T r).$$

The first weight vector is:

$$v_1 = U_1,$$

and the first score is:

$$s_1 = X_{\text{std}}v_1.$$

Subsequent components are computed by deflating X_{std} and y:

$$X_{\text{std}} \leftarrow X_{\text{std}} - s_1 p_1^T, \quad y \leftarrow y - s_1 q_1.$$

This process is repeated to obtain k latent components.

- Regression on latent components

  Once the latent components are extracted, we estimate the regression:

  $$r = S\beta + \varepsilon.$$

  The least squares estimate is:

  $$\hat{\beta} = (S^T S)^{-1} S^T r.$$

  The final prediction is:

  $$\hat{r} = S\hat{\beta} = X_{\text{std}} V (S^T S)^{-1} S^T r.$$

Appendix D. Details: Random forest (RF) and Gradient Boosted Regression Trees (GBRT)

Given the dataset $D = \{(X_{it}, r_{it})\}_{i=1,\dots N}^{t=1,\dots T}$ with input features $X_{it} \in \mathbb{R}^p$ and response $r_{it} \in \mathbb{R}$, RF builds B individual trees, each trained on a bootstrapped subset $D_b \subset D$. Each individual tree in a Random Forest is constructed using the following steps:

- Select a bootstrap sample D_b of size NT (sampling with replacement). Each tree is trained on a different subset of the training data.
- At each split in the tree, randomly sample a subset of m features ($m < p$) instead of considering all p features. This random selecting reduces overfitting and encourages diversity among trees.
- For each selected feature subset, determine the optimal split by maximizing an impurity reduction criterion such as variance reduction (for regression) (Breiman, 2017). For regression trees, the split criterion minimizes variance. Given a dataset with response values r_i, variance is calculated as:

  $$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2,$$

  where $\bar{r}$ is the mean of the node. The optimal split minimizes the weighted variance of the left and right nodes:

  $$\sigma^2_{\text{split}} = \frac{N_L}{N}\sigma^2_L + \frac{N_R}{N}\sigma^2_R.$$

- Grow the tree recursively by applying the same splitting strategy to each child node until a stopping criterion is met (Breiman, 2017), such as minimum number of samples required to split a node, maximum depth limit to control complexity, leaf nodes reaching a minimal sample size.
- Each tree provides a prediction by averaging the outcomes in each leaf node:

  $$f_b(x) = \sum_{l=1}^{L_b} c_{bl}\mathbb{I}(x \in R_{bl})$$

  where L_b is the number of leaf nodes, R_{bl} represents the regions of the feature space assigned to each leaf, and c_{bl} are the leaf node predictions.

- The final prediction for Random Forest is the aggregation of individual tree predictions, which is typically computed as an average for regression:

$$\hat{r} = \frac{1}{B}\sum_{b=1}^{B} f_b(x).$$

In the process of our model fitting, the relevant hyper parameters: for booststrapping our number of trees is 300, our stopping criterion is when the maximum tree depth reaches 4. Generally speaking, a shallow tree will have less risk of overfitting and improve generalization capabilities.

Given training data $D = \{(X_{it}, r_{it})\}_{i=1,\ldots N}^{t=1,,,T}$, GBRT approximates the target variable $F(x)$ by iteratively constructing trees:

- Initialize the model with a constant value (often the mean response):

$$F_0(x) = \arg\min_c \sum_{i=1}^{N}\sum_{t=1}^{T} L(r_{it}, c).$$

- For $b = 1, 2, \ldots, B$:
  - Compute the residuals (negative gradient of loss function L):

$$res_{itb} = -\left.\frac{\partial L(r_{it}, F(x_{it}))}{\partial F(x_{it})}\right|_{F(x_{it})=F_{b-1}(x_{it})}.$$

  - Fit a regression tree $h_b(x)$ to predict the residuals res_{itb}.
  - Compute the step size γ_b by solving:

$$\gamma_b = \arg\min_{\gamma} \sum_{i=1}^{N}\sum_{t=1}^{T} L(r_{it}, F_{b-1}(x_{it}) + \gamma h_b(x_{it})).$$

  - Update the model:

$$F_b(x) = F_{b-1}(x) + \gamma_b h_b(x).$$

- The final prediction is given by:

$$\hat{r} = F_B(x) = \sum_{b=1}^{B} \gamma_b h_b(x).$$