Beyond the Black Box: *An Intuitive Approach to Investment Prediction with Machine Learning*

YIMOU LI, DAVID TURKINGTON, AND ALIREZA YAZDANI

YIMOU LI
is assistant vice president and machine learning researcher at State Street Associates in Cambridge, MA.
yli24@statestreet.com

DAVID TURKINGTON
is senior managing director and head of Portfolio and Risk Research at State Street Associates in Cambridge, MA.
dturkington@statestreet.com

ALIREZA YAZDANI
is vice president and machine learning research manager at State Street Associates in Cambridge, MA.
ayazdani@statestreet.com

KEY FINDINGS

- This article presents a framework for the implementation and interpretation of machine learning model predictions applied to investment portfolios.
- Model predictions are decomposed into the linear, nonlinear, and interaction components, and their predictive efficacy is evaluated using these components.
- Using a currency prediction case study, it is demonstrated that machine learning models reliably identify known effects and find new nonlinear relationships and interactions.

ABSTRACT: *The complexity of machine learning models presents a substantial barrier to their adoption for many investors. The algorithms that generate machine learning predictions are sometimes regarded as a black box and demand interpretation. In this article, the authors present a framework for demystifying the behavior of machine learning models. They decompose model predictions into linear, nonlinear, and interaction components and study a model's predictive efficacy using the same components. Together, this forms a fingerprint to summarize key characteristics, similarities, and differences among different models. The presented framework is demonstrated in a case study applying random forest, gradient boosting machine, and neural network models to the challenge of predicting monthly currency returns. All models reliably identify intuitive effects in the currency market but also find new relationships attributable to nonlinearities and variable interactions. The authors argue that an understanding of these predictive components may help astute investors generate superior risk-adjusted returns.*

TOPICS: ***Statistical methods, simulations, big data/machine learning****

Machine learning has led to impressive results in many fields. Although the specific applications and types of models vary widely, they generally owe their success to greater computational efficiency paired with models that are less dependent on simplifying assumptions, such as stylized forms of probability distributions, than those in the past. As a result, sophisticated machine learning models have the ability to capture nonlinear dependencies and interaction effects that may lead to superior predictions. On the other hand, the inherent complexity of these models creates challenges for interpretation and understanding. This issue is especially relevant to investment applications. Predicting time-series returns in

*All articles are now categorized by topics and subtopics. **View at PM-Research.com.**

financial markets is fundamentally different from other mainstream applications of machine learning, such as image recognition, where the underlying data generation process is relatively stable over time. In contrast, the behavior of financial markets is ruled by constant change and uncertainty as a result of competitive dynamics and structural shifts. This means that data from more distant history may be less relevant for training machine learning algorithms, and we are left with an inherently short data sample and a low signal-to-noise ratio. Because of such unique characteristics of financial information, some practitioners (López de Prado 2019; Simonian and Fabozzi 2019) have called for establishing financial data science as a standalone field in its own right, wherein greater emphasis is placed on empiricism and data-driven expansions of traditional financial econometrics.

The two primary goals of data analysis, as noted by Breiman (2001), are to make a prediction and to obtain information that aids in understanding. Along these lines, we pose two distinct questions:

1. Can machine learning algorithms detect patterns in financial data that lead to superior investment returns?
2. How do the algorithms process the data to form predictions?

In this article we attempt to answer these questions, particularly by proposing a methodology to address the second question, which is often neglected in the literature. Specifically, we propose a set of interpretability metrics, collectively named a *model fingerprint*, to decompose the behavior of any model's predictions into linear, nonlinear, and interaction effects among predictors. We also show how to decompose the model's predictive efficacy into these components. Next, we explore these concepts in the context of foreign currency investing. We present a case study applying random forest, gradient boosting machine, and neural network algorithms to predict one-month-forward currency returns. We choose to illustrate these concepts in the currency market for a number of important reasons. The currency market is one of the largest and most actively traded global markets and thus a very important one for many investors. In particular, we study the behavior of a subset of exchange rate investments, consisting of all pairs (cross-rates) of the 10 largest currencies. Though the amount of data involved in currency prediction is not necessarily large, the problem is quite complex because of the many economic effects involved. Another motivation for this case study is that, although there is a rich body of research in economics and finance to motivate the choice of predictor variables and provide helpful intuition, many traditional quantitative strategies have failed to deliver reliable results in the aftermath of the 2008 financial crisis (Czasonis, Pamir, and Turkington 2019). There is a practical need for improvement.

Previous research has applied machine learning to investment prediction, with encouraging results. Many such studies, however, have focused on security selection within the equity market. For example, Heaton, Polson, and Witte (2016) explored the use of deep learning models for financial prediction problems, including pricing securities, constructing portfolios, and risk management. Gu, Kelly, and Xiu (2019) showed that the cross section of US stock returns can be predicted well with machine learning models, and neural networks in particular. Rasekhschaffe and Jones (2019) explored machine learning for stock selection and forecasting the cross section of stock returns. We extend this literature on empirical findings by offering promising results for currency market predictions.

Regarding model interpretation, theoretical research has offered a variety of ways to study the information processing mechanisms of machine learning algorithms. Molnar (2019) provided a useful survey of existing approaches to interpretability, including ways to quantify the influence of a given predictor toward the model outcome, as well as ways to analyze the nature of the relationship (e.g., linear, nonlinear) between predictors and outcome. For instance, it is common to compute a measure of variable importance to quantify the predictive strength of each input variable in a model, but the method for doing so is usually specific to the model in question, limited in scope, and far from unified. For example, the importance of a predictor in a multiple linear regression might be defined as the absolute value of its t-statistic. For tree-based models, however, the total reduction in prediction error over all nodes that select the variable of interest is commonly used. The list extends with many proposals of custom variable importance scores for specific model types (Kuhn 2008). A methodology known as partial dependency (Friedman 2001) can be used to understand the relationship between predictors and the model outcome. Another measure proposed by Greenwell, Boehmke, and McCarthy (2018) uses the

flatness of a predictor's partial dependency as a measure of variable influence.

Our proposed approach, which we call a model fingerprint, is distinguished from those in the earlier literature. It decomposes model predictions into linear, nonlinear, and interaction components and offers an intuitive, model-independent notion of predictive efficacy. The fingerprint metrics are expressed directly in units of the predicted returns, making them comparable across any set of models. This approach offers valuable insights in understanding how a machine learning model outcome is affected by the presence of interactions among different drivers of performance.

We structure the remainder of the article as follows. First, we introduce our methodology for model fingerprints, including a decomposition of a model's predictions and a decomposition of its performance. Second, we present an intuitive application to currency investing, including an evaluation of performance on both training and testing samples. Last, we summarize and conclude.

METHODOLOGY

Machine Learning Models

The methodology we present for model fingerprints is general and applies to any predictive model, but it is helpful to keep in mind some specific examples. In this paper, we consider three machine learning models. Even with just three models, this collection has diversity in terms of model characteristics and capabilities, architecture complexity (e.g., nonlinear, tree based), and learning styles. We briefly describe the three models and refer the interested reader to further machine learning resources, such as work by Kuhn and Johnson (2013) and Hastie, Tibshirani, and Friedman (2008), for details.

1. **Random forests** aggregate the outcomes of many simple decision trees developed independently on randomly selected subsets of predictors and data. This process, known as bootstrap aggregating or *bagging* (Hastie, Tibshirani, and Friedman 2008), aims to preserve the ability of tree-based models to capture conditional effects in data while mitigating their tendency to overfit the training sample.
2. **Gradient boosting machines** also use simple decision trees as base learners, but they use an additive model to minimize prediction errors (given a specified loss function) and proceed iteratively to fit the residuals from previous iterations, leading to a phenomenon known as *boosting* (Friedman 2001).
3. **Neural networks** consist of nested data processing layers transforming inputs into predictions. In deep networks, multiple nodes and hidden layers provide the capacity to model highly complex relationships (Goodfellow, Bengio, and Courville 2016). In our application to currencies, we use a relatively shallow feed-forward network architecture with no more than four hidden layers.

Model Fingerprints (Part 1): Decomposition of Predictions

After a model is selected and calibrated on training data, it can be used to map any desired set of input values into a predicted value. Although it is usually straightforward to calculate the prediction value, the actual prediction mechanism can be quite intricate and difficult to visualize or understand for all but the simplest cases. Our goal is to summarize the characteristics of a given model in terms of its linear, nonlinear, and interaction effects. In particular, we quantify how much variation in predicted values results from variation in each input variable—and each pair of variables—in isolation, holding all else constant. We refer to this set of metrics as a fingerprint because it provides a concise and distinctive description of the predictive characteristics of the calibrated model.

Our methodology modifies and extends the notion of partial dependence introduced by Friedman (2001). The partial dependence function captures the marginal prediction derived from the average effect of one variable in isolation. Let us denote a model prediction function as

$$\hat{y} = \hat{f}(x_1, x_2, \ldots, x_m) \tag{1}$$

This prediction depends on each of the M input variables, whereas the partial dependence function only depends on one of the input variables, x_k. For a given value of x_k, this partial dependence function returns the expected value of the prediction over all other possible values for the other predictors, which we denote as $x_{\setminus k}$:

$$\hat{y}_k = \hat{f}_k(x_k) = E_{x_{\setminus k}}[\hat{f}(x_1, x_2, \ldots, x_m)] = \int \hat{f}(x_k, x_{\setminus k}) p(x_{\setminus k}) dx_{\setminus k} \tag{2}$$

By marginalizing the prediction output over the distribution of all other predictor variables, the partial dependence function provides an intuitive sense for the marginal impact of the variable of interest, which we may think of as a partial prediction. In practice, the procedure to estimate the partial dependence function from the empirical data is as follows:

1. Choose a permissible value for x_k.
2. Combine this value with one of the actual input vectors for the remaining variables, $x_{\setminus k}$, and generate a new prediction from the function: $\hat{y} = \hat{f}(x_1, x_2, \ldots, x_m)$.
3. Repeat step 2 with every input vector for $x_{\setminus k}$, holding the value for x_k constant, and record all predictions.
4. Average all the predictions for this value of x_k to arrive at the value of the partial prediction at that point, $\hat{y}_{x_k}$.
5. Repeat steps 1 through 4 for any desired values of x_k and plot the resulting function.

The partial dependence function will have small deviations if a given variable has little influence on the model's predictions. Alternatively, if the variable is highly influential, we will observe large fluctuations in prediction based on changing the input values. When this procedure is applied to an ordinary linear regression model, the plot will be a straight line with a slope equal to the regression coefficient of x_k. Therefore, it is intuitive to view the partial dependence function as a generalized version of a regression coefficient that allows for nonlinear effects.

Next, we decompose a variable's marginal impact into a linear component and a nonlinear component by obtaining the best fit (least squares) regression line for the partial dependence function. We define the linear prediction effect—the predictive contribution of the linear component—as the mean absolute deviation of the linear predictions around their average value.

$$\textit{Linear prediction effect}\ (x_k) = \frac{1}{N}\sum_{i=1}^{N} abs\left(\hat{l}_k[x_{k,i}] - \frac{1}{N}\sum_{j=1}^{N}\hat{f}_k[x_{k,j}]\right) \quad (3)$$

In Equation 3, for a given predictor x_k, the prediction $\hat{l}_k(x_{k,i})$ results from the linear least square fit of its partial dependence function, and $x_{k,i}$ is the ith value of x_k in the dataset.

Next, we define the nonlinear prediction effect—the predictive contribution of the nonlinear component—as the mean absolute deviation of the total marginal (single variable) effect around its corresponding linear effect. When this procedure is applied to an ordinary linear model, the nonlinear effects equal precisely zero, as they should.

$$\textit{Nonlinear prediction effect}\,(x_k) = \frac{1}{N}\sum_{i=1}^{N} abs(\hat{f}_k[x_{k,i}] - \hat{l}_k[(x_{k,i})]) \quad (4)$$

Exhibit 1 depicts these relationships graphically. The linear and nonlinear effects are intuitively related to the shaded areas, as shown in Exhibit 1.[1]

A similar method can be applied to isolate the interaction effects attributable to pairs of variables x_k and x_l, simultaneously. The procedure for doing this is the same as given earlier, but in step 1 values for both variables are chosen jointly.

$$\hat{y}_{k,l} = \hat{f}_{k,l}(x_k, x_l) = E_{x_k}\{E_{x_l}[\hat{f}(x_1, x_2, \ldots, x_m)]\} \quad (5)$$

We define the *pairwise interaction effect* as the de-meaned joint partial prediction of the two variables minus the de-meaned partial predictions of each variable independently. When this procedure is applied to an ordinary linear model, the interaction effects equal precisely zero, as they should.

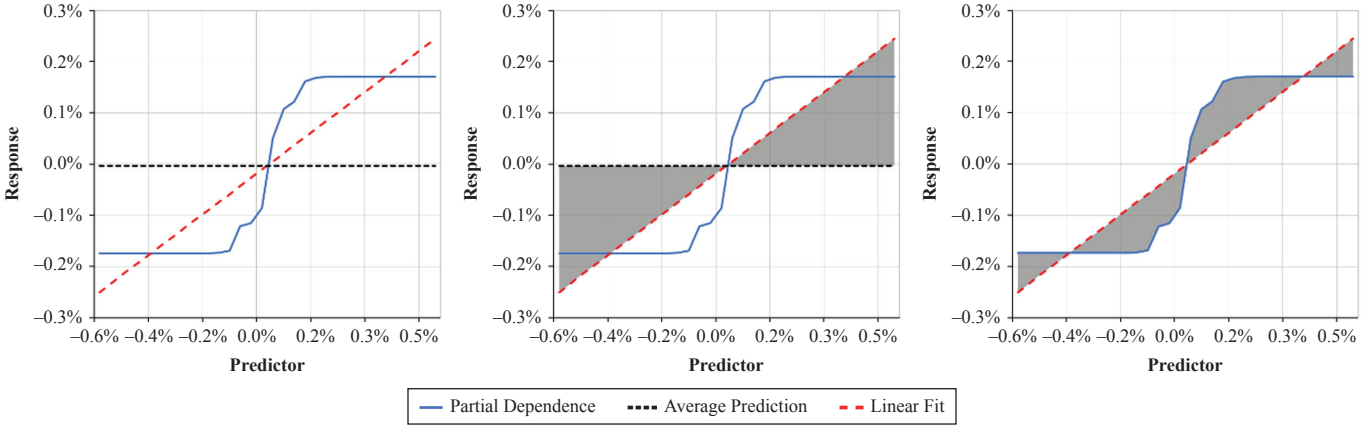
$$\textit{Pairwise interaction effect}\,(x_k, x_l) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} abs[\hat{f}_{k,l}(x_{k,i}, x_{l,j}) - \hat{f}_k(x_{k,i}) - \hat{f}_l(x_{l,j})] \quad (6)$$

Our approach to defining the pairwise interaction effect is conceptually similar to the H-statistic introduced by Friedman and Popescu (2008). The H-statistic compares joint variable interactions to the sum of the relevant individual variable effects, all measured with par-

[1] This illustration is based on the random forest model, which will be discussed in the next section. The area under the curve is a stylized example that applies exactly if the predictor values are uniformly distributed across their domain. In practice, we sum the absolute deviations over all observed values for the predictor, so some parts of the predictive function will be more highly represented than others.

EXHIBIT 1

Partial Prediction (left), Linear Effect (middle), and Nonlinear Effect (right)



tial dependence functions. It sums the squares of these incremental interaction effects across every data point and then divides by the sum of the squared total joint predictions. In other words, it equals the variance of incremental interaction effects divided by the variance of the total joint predictions. Our approach in Equation 6 differs in two ways. First, we use the mean absolute deviation to measure the extent of the effect, rather than the squared deviations (variance), which makes our measure less sensitive to outliers in the data. Second, we explicitly keep our measure in units of the predicted variable for easy interpretation and comparison to the linear and nonlinear prediction effects we measured previously, so we present the mean absolute deviation directly, without the denominator (normalization) that is included in the H-statistic.

Exhibit 2 shows an example of an isolated pairwise interaction effect in two dimensions. The total extent of the interaction effect is intuitively related to the volume under the surface defined by these two dimensions (analogous to the shaded areas in Exhibit 1).

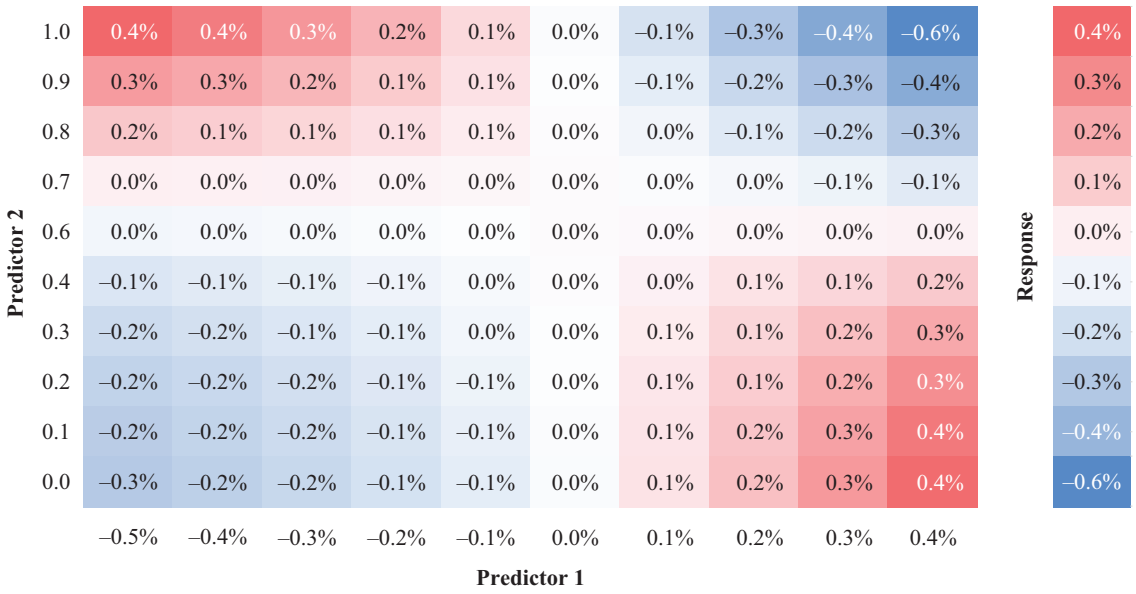
The metrics we have described here offer attractive properties. First, they can be applied to any set of predictive models, and fingerprint results are comparable across them. Second, they measure linear, nonlinear, and interaction effects in common units that are economically meaningful (units of the response variable that is being predicted). Third, they extend highly intuitive and familiar ordinary linear regression concepts to the machine learning models. In summary, they help demystify the drivers of model predictions and facilitate interpretations of why a model behaves the way it does.

Model Fingerprints (Part 2): Decomposition of Performance

Although the fingerprint metrics from Part 1 provide insight into the behavior of a predictive function, they do not provide any information about the effectiveness of those predictions. We now turn our attention to this issue and apply the same framework to decompose prediction efficacy into its component parts. We choose to measure efficacy in terms of the performance of portfolios formed from the model's predictions. This way, the assessment is made in economically meaningful units, is diversified across assets (such as currency pairs) at each point in time to mitigate noise, and can be observed as a time series for additional insights.

We have already described the methodology to decompose the overall prediction function. To attribute a model's performance to its component parts, we extract partial predictions based only on a subset of the predictive components and form portfolios from those partial predictions. In Part 1, we discussed partial predictions based on the information from one input variable. The partial predictions that we consider now are aggregated across all of the predictor variables in the model, but they are partial in the sense that they only use a subset of the predictive components from our decomposition (linear, nonlinear, and interaction effects).

EXHIBIT 2
Pairwise Interaction Effect



$$\hat{y} = \sum_{k=1}^{M} Linear\ effect(x_k) + \sum_{k=1}^{M} Nonlinear\ effect(x_k) + \sum_{\{k,l\}\in M, k\neq l} Pairwise\ interaction\ effect(x_k, x_l) + Higher\text{-}order\ interaction\ effects \quad (7)$$

We compare the returns of portfolios built from different subsets of predictive components to arrive at time series of returns corresponding to each component.[2] In particular, we define each component as follows:

1. The **linear *(unconditional)* performance effect** is equal to the time series of returns of portfolios formed from linear predictions in isolation.
2. The **pairwise interactions (*conditional*) performance effect** is equal to the time series of returns of portfolios formed from the combined linear and pairwise interaction predictions minus the time series of returns from step 1.
3. The **nonlinear (*sizing*) performance effect** is equal to the time series of returns of portfolios formed from the combined linear, nonlinear, and pairwise interaction predictions minus the time series of returns from steps 1 and 2.
4. The **higher-order interactions performance effect** is equal to the time series of returns of portfolios formed from the full predictive model minus the time series of returns from steps 1, 2, and 3. This captures the influence of higher-order interactions that occur above and beyond the pairwise interactions on the predictive models.[3]

The sequence in which we compute these performance effects may at first seem strange, but it serves an important purpose. The interaction effects come second in the sequence because they are, in a sense, more fundamentally important to investment prediction than are the nonlinear sizing effects. Interactions allow for conditional relationships between variables, and conditional relationships can dramatically change the directionality of a prediction. For example, suppose that variable A is a positive predictor when B is low but a negative predictor when B is high. Furthermore, imagine that variable A is a stronger predictor when its value is at extreme highs or lows than when it is

[2] In our empirical analysis, we use a simple ranking and equal weighting to construct portfolios, but other portfolio construction methods could be used instead.

[3] We quantify the impact of higher-order interaction effects by exclusion because they are too numerous to evaluate directly.

in the middle of its range (a nonlinear effect). In this example, trying to isolate the nonlinear performance effect of A will produce very counterintuitive—and possibly meaningless—results if we do not account for its conditionality on B. The nonlinear performance in isolation implies a larger prediction for both high and low values of A (imagine a U-shaped curve). Only in combination with B do we see that half of these strong positions take the opposite sign when they enter into the full model's prediction. Although it may not be possible to prevent this type of issue altogether, we suspect that for many popular models, considering interaction effects before nonlinear effects for performance decomposition is likely to yield a more useful interpretive analysis. In summary, it seems reasonable to consider conditional relationships before nonlinear sizing effects. Lastly, it is worth noting that this issue does not affect the decomposition of predictions in the previous section because that analysis is concerned with the magnitude of predictive components and not their positive or negative direction.

EMPIRICAL APPLICATION TO FOREIGN CURRENCY INVESTING

In this section, we apply the three machine learning techniques described earlier to the task of foreign currency prediction. This real-world empirical study allows us to identify intuitive relationships using the model fingerprint approach. We begin by describing the currency model specification, data, and procedure for training. It is critical to distinguish between performance in the training sample (before 2016) and performance in the testing sample (after 2016). First, we thoroughly examine the model fingerprints for prediction and performance based on the training data to better understand the behavior and data processing mechanisms of each model. Second, we present performance results and interpretation for the testing sample, which provides a realistic indication of how the models behave when exposed to previously unseen data.

Model Specification and Training

The goal of our empirical study is to predict one-month-forward returns for major currencies. We focus on the total return of forward contracts because they represent investable exposures in the currency market.[4] We form a dataset of monthly returns for each of the exchange rate pairs from the G10 currencies: Australian dollar (AUD), Canadian dollar (CAD), Swiss franc (CHF), Euro (EUR),[5] British pound (GBP), Japanese yen (JPY), Norwegian krone (NOK), New Zealand dollar (NZD), Swedish krona (SEK), and US dollar (USD). Our full dataset includes the returns of each of the 90 currency pairs (quoted in both directions, to avoid the arbitrary effect of one quoting convention) observed for 351 months from January 1990 to March 2019, for a total of 31,950 observations. We split the data into a training sample spanning January 1990 to December 2015 and a test sample spanning January 2016 to March 2019, which we reserve for final performance evaluation.

We structure the prediction problem as a panel regression: the return of a given currency pair at a given point in time is to be predicted with the information available about that pair. The panel regression combines cross-sectional and time-series information, allowing the model to be trained on increased variability from a greater number of observations.[6] We deliberately restrict our attention to a narrow set of established currency factors as predictors. This simplicity makes it easier to view the similarities and differences in how each model processes the data to form predictions. Each predictor is motivated by established results in the currency market:

1. The **short-term interest rate differential** between countries forms the basis for the carry trade, where forward contracts for currencies with higher interest rates have historically outperformed those with lower interest rates (Burnside, Eichenbaum, and Rebelo 2011).

[4] A currency forward contract is an agreement to buy or sell a given currency versus another currency at some point in the future. The market price of the forward rate is determined by a no-arbitrage condition called Covered Interest Parity because one may achieve the same payout as the forward contract by borrowing money in one currency and investing it in the other. Thus, a currency position—whether implemented using a forward contract or by borrowing and investing across countries—is self-funding and has a total return equal to the interest rate differential between the two countries plus the change in the spot exchange rate over the investment horizon. We model actual forward prices, which are investable.

[5] Before the introduction of the Euro, we proxy it with the German mark.

[6] The information about a given currency pair partly overlaps with the information about other pairs that contain one of the same currencies. However, the information is not completely redundant and therefore helps in model training.

2. The **trailing five-year spot return adjusted for trailing five-year inflation differential** provides an indication of the deviation from fair value for a currency pair, based on the notion of relative purchasing power parity, and underpins the valuation trade. Currencies that are undervalued have historically outperformed those that are overvalued (Czasonis, Pamir, and Turkington 2019).
3. The **trailing one-year spot return** for a currency pair informs a trend strategy. Currencies that recently rose in value have historically outperformed those that recently lost value (Burnside, Eichenbaum, and Rebelo 2011).
4. The **trailing one-year equity return differential** between countries constitutes an equity differential trade. Currencies whose local equity market has risen recently have historically outperformed those whose equity market has done relatively poorly (Turkington and Yazdani 2019).
5. **Currency market turbulence** is a multivariate unusualness score of the currency market based on the Mahalanobis distance (Kritzman and Li 2010), and we employ it as a measure of the overall market risk profile that is more persistent than the volatility of individual currency pairs. Larger magnitude return swings and correlation breakdowns both increase the amount of turbulence observed. Some currency factors, such as the carry trade, have historically behaved very differently during turbulent versus quiet periods.

Hyperparameter tuning is an important issue that affects the training of machine learning models. Hyperparameters differ by model, but examples include the number and depth of decision trees in a random forest, the depth and sample rate of trees in a gradient boosting machine, and the activation function and size of hidden layers in a neural network. It is often desirable for these parameter choices to be informed directly by the data. The goal of proper calibration is to achieve (near) optimal performance of a model for the task at hand, while minimizing the risk of overfitting in the training sample. As such, hyperparameter tuning can be challenging and a task that requires an extra layer of data processing. To this end, we use a 10-fold cross-validation approach to identify the most desirable set of hyperparameters while mitigating the risk of overfitting to the data. In essence, cross validation creates synthetic unseen evaluation samples from subsets of the training data. Specifically, we divide our panel of training data into 10 contiguous and nonoverlapping blocks of time. For a given set of hyperparameters, we fit the model on every combination of nine blocks and evaluate model performance using root mean squared error (RMSE) on the remaining block. We store the composite predictive performance (RMSE across all evaluation blocks) of the model under the current set of hyperparameters and then search for the hyperparameters that result in the best overall fit.

The overall performance of machine learning regression models is typically evaluated using measures such as R^2 and RMSE. Although these measures can be informative, they may not adequately reflect how a model performs in a financial portfolio setting. Thus, it is prudent to test the return and risk performance of realistic portfolios formed from model predictions. To do this, we identify for each month the 27 currency pairs with the largest prediction magnitude and assign long or short positions depending on each prediction's directional sign. The selection of 27 out of the 45 nonoverlapping pairs ensures a diversified portfolio by avoiding significant exposure to one single currency and is in line with a traditional top three, bottom three approach to building long–short currency portfolios.

Model Fingerprints: Evaluation on the Training Sample

Exhibit 3 presents the model fingerprints of predictor influence. It is notable that the relative size of linear effects is nearly identical across all three models (even though their absolute size differs). All models exhibit the most nonlinearity with respect to the interest rate differential factor but differ in their other nonlinear effects. The interaction between currency turbulence and interest rate differential is deemed the most salient pairwise interaction effect in each case, but again the other interaction effects vary across the models. A closer look into the interaction heatmaps shown in Exhibit 4 reveals that all three models make predictions in line with the conventional carry trade (based on the interest rate differential) when turbulence is low. However, during highly turbulent regimes, all three models reverse this relationship. This interaction effect aligns with previous

EXHIBIT 3

Model Fingerprints (decomposition of predictions)

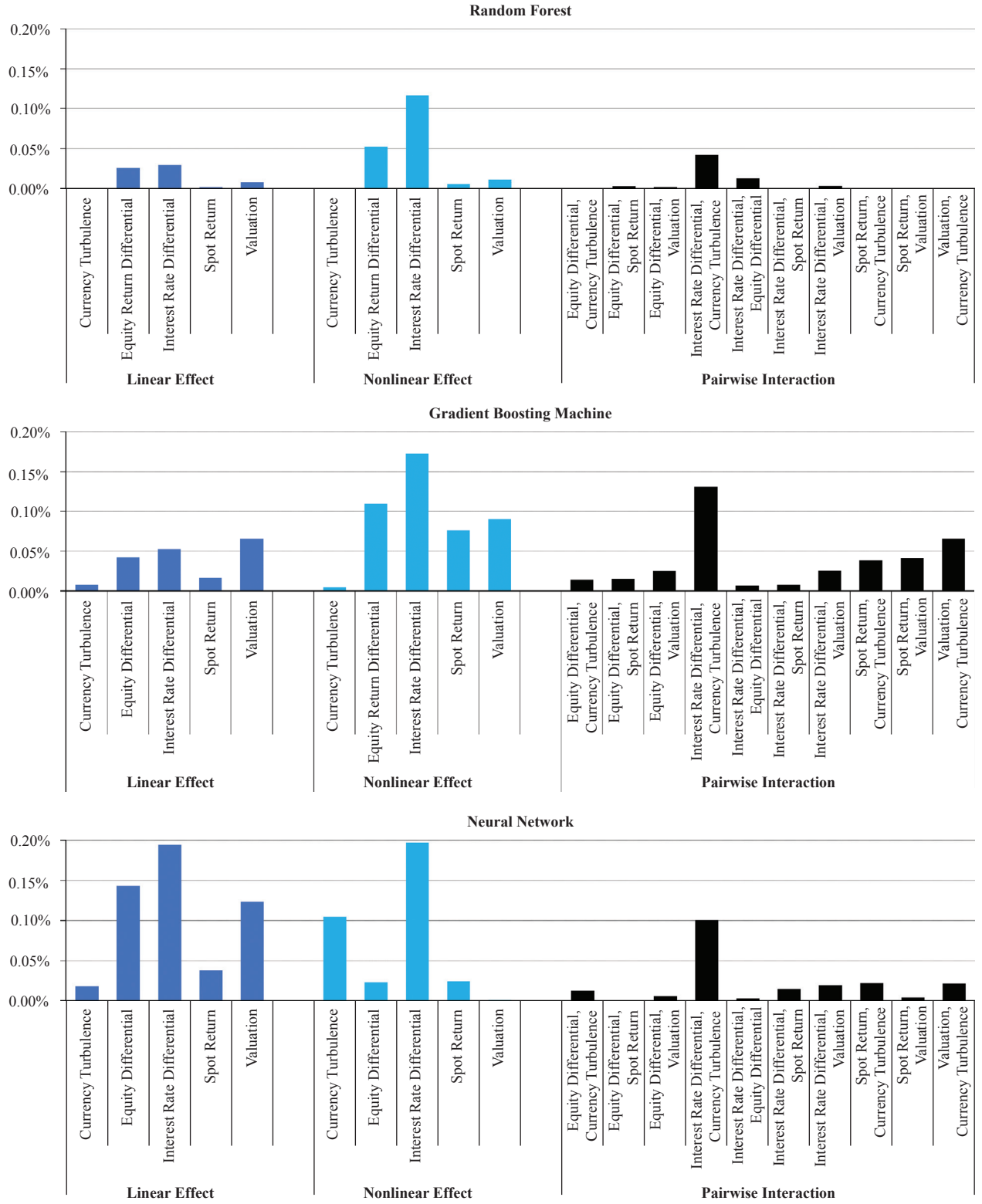
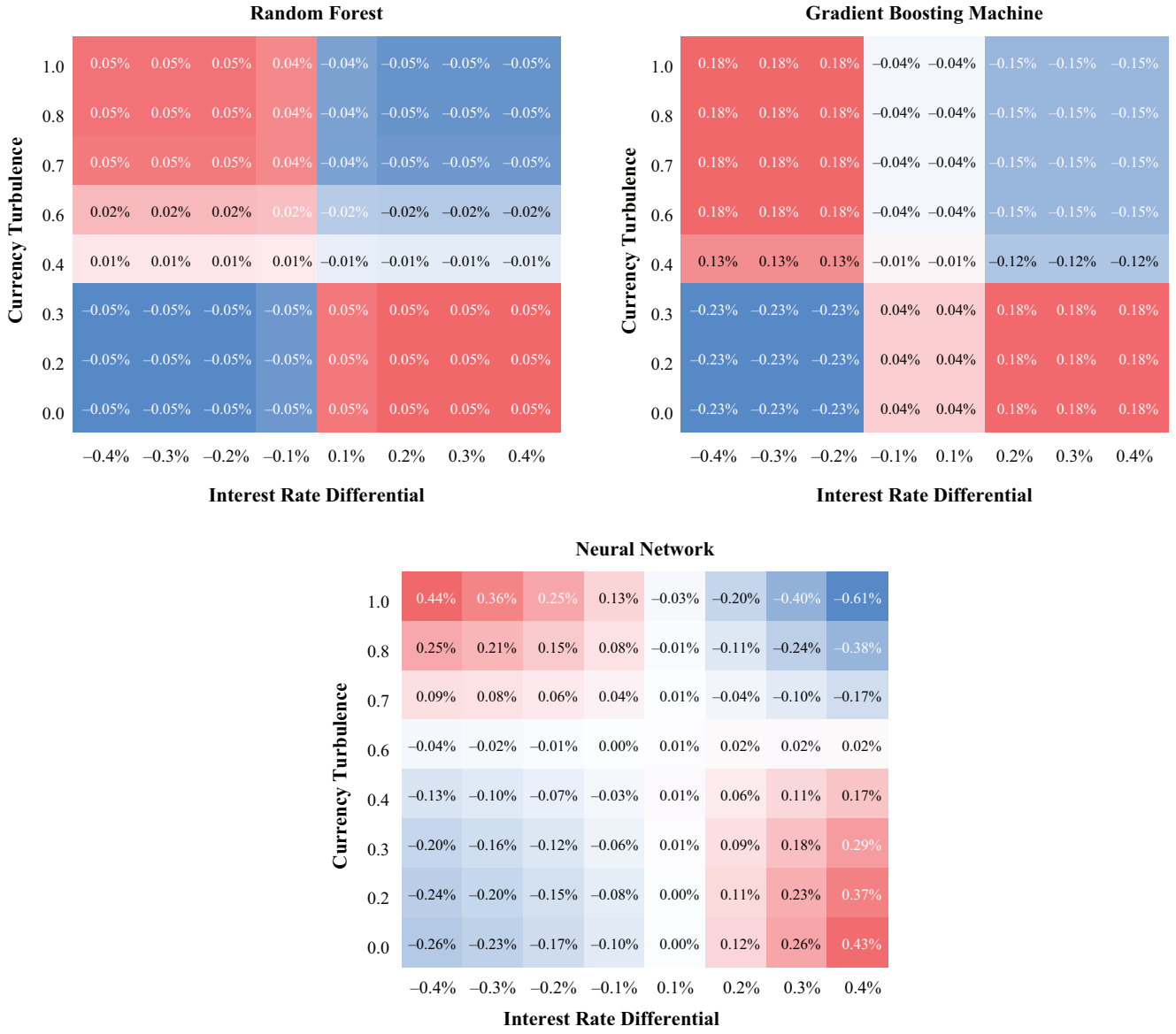


EXHIBIT 4

Interaction Effects between Interest Rate Differential and Currency Turbulence

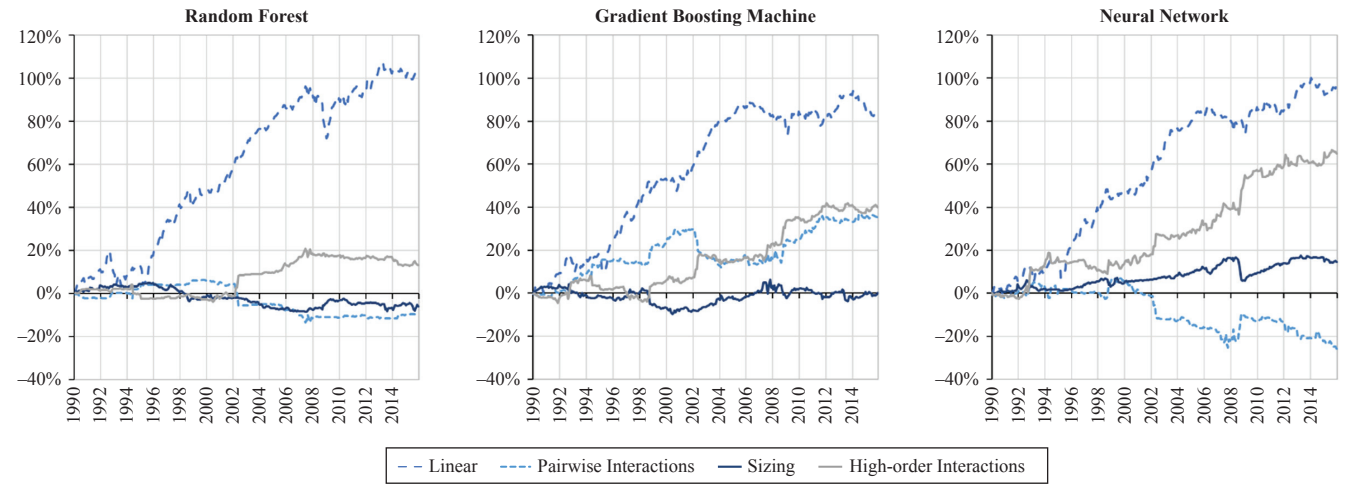


research on how turbulent regimes affect the carry trade (Kritzman and Li 2010).

Exhibit 5 presents the model fingerprints from the perspective of predictive efficacy. Pairwise interactions suggest conditional relationships at each point in time. Higher-order interactions performing in the same direction as the pairwise (as is often the case in 2008, for example) indicates that the higher-order conditional effects are confirming and even amplifying the same understanding as the pairwise effects. On the other hand, there are times when the two move in opposition, which indicates that the higher-order interaction effects are neutralizing or reversing the pairwise effect based on the confounding influence of other conditions that prevail at that time. In fact, all three models exhibit a large loss in 2002 that would have occurred from pairwise interactions but was overturned by a higher-order consideration in all three cases. Overall, the performance

EXHIBIT 5

Model Fingerprints (decomposition of performance)



decomposition provides insights into the degree to which a model relies on the component prediction effects.

Performance in the Training Sample

We are particularly interested in whether machine learning strategies exceeded the performance of traditional currency strategies and a benchmark linear regression model. A key consideration when training machine learning models is to reduce overfitting by avoiding aggressive data mining. To this end, we followed conventional practice in structuring the training and validation process to minimize the expected gap in performance in the training versus testing sample. Of course, we must still acknowledge that even with the cross-validation process, which mitigates the potential to overfit the data, these models have been selected for their performance on the training sample and have therefore benefited from learning some of the features of the training data on which we are evaluating performance here. Even with this caveat, a thorough analysis of training sample performance is important to gain an understanding of what each model is thinking. In the next section, we will evaluate performance in the testing sample.

Exhibits 6 and 7 present training sample performance across all models, including portfolios that are formed on only one predictive variable in isolation (carry, trend, valuation, and equity differential). In terms of risk-adjusted return (information ratio), machine learning models outperformed the linear model, which in turn outperformed the simple traditional strategies. Grading boosting had the best in-sample performance, with the highest annualized returns and one of the lowest levels of risk. At the same time, returns from the gradient boosting machine had the highest excess kurtosis, indicating a propensity for occasionally extreme monthly returns. It is also interesting to note the strong similarity in returns for the gradient boosting and neural network models. Both outperform the other models by a significant margin, which perhaps is to be expected given the nonlinear and interaction effects they find.

Performance in the Testing Sample

The understanding we have gained so far allows us to make interpretations about model tendencies, similarities, differences, and performance in the training sample. The performance reliability of a model is reflected not only in its training sample performance but also when evaluated on unseen test data. Exhibits 8 and 9 present the results for the testing sample. As in the training sample, gradient boosting machine performs well, continuing to generate comparatively high returns and low risk. However, the overall performance gap with other models is not as wide, which may indicate a mild degree of (inevitable) overfitting by the

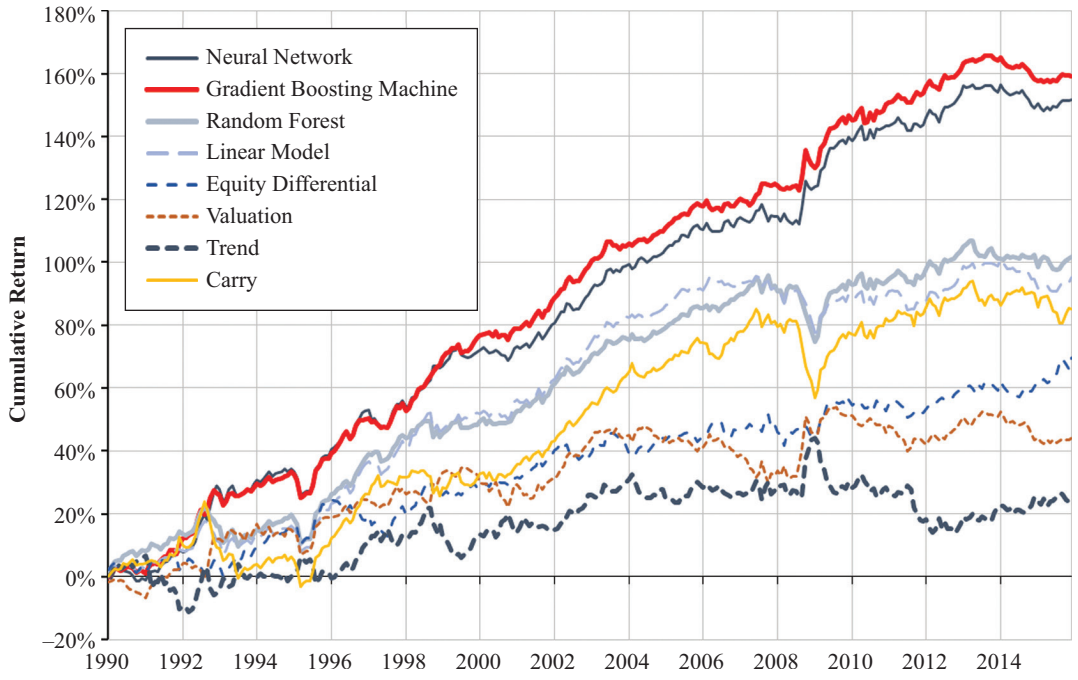
EXHIBIT 6

Portfolio Performance Summary Statistics (training data)

	Carry	Trend	Valuation	Equity Differential	Linear Model	Random Forest	Gradient Boosting Machine	Neural Network
Return	3.27%	0.91%	1.70%	2.68%	3.66%	3.92%	6.12%	5.83%
Risk	6.50%	6.75%	5.82%	5.00%	5.22%	5.40%	4.94%	5.41%
Ratio	0.50	0.13	0.29	0.54	0.70	0.73	1.24	1.08
Skewness	–0.69	0.10	0.99	–0.26	–0.60	–0.73	0.32	0.25
Kurtosis	1.24	3.97	4.89	0.84	2.02	2.21	4.93	3.16
Hit Rate	0.63	0.52	0.51	0.57	0.63	0.66	0.68	0.67

EXHIBIT 7

Portfolio Cumulative Returns for Different Strategies (training data)



gradient-boosting model. Overall, there is a convergence in performance during this sample, with little separation across the linear model and machine learning models. Most of the traditional currency strategies underperformed, with the exception of the equity differential. This may indicate a reduced opportunity set based on the predictor variables we have included.

Exhibit 10 shows the performance decomposition of the machine learning models over the testing sample and supports a similar conclusion. Again, we stress that that our goal in this paper is to understand the tendencies of each model and to be able to demystify the components of performance by attributing them to their component parts. We have intentionally erred on the side of simplicity and have not tried to build the best possible model. Exploring a wider range of variables and models to enhance performance further would be an interesting and useful extension of our case study for currency investing.

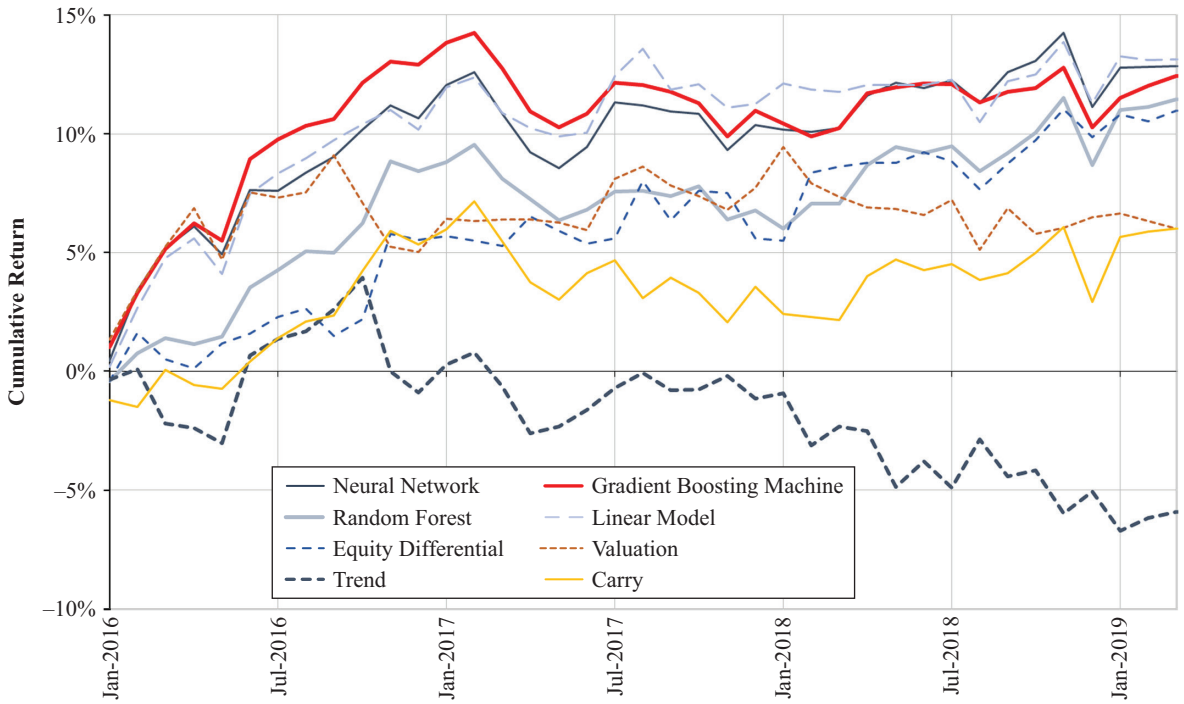
EXHIBIT 8

Portfolio Performance Summary Statistics (test data)

	Carry	Trend	Valuation	Equity Differential	Linear Model	Random Forest	Gradient Boosting Machine	Neural Network
Return	1.85%	–1.82%	1.85%	3.37%	4.04%	3.52%	3.82%	3.96%
Risk	4.14%	4.84%	4.23%	3.98%	4.31%	3.66%	3.88%	4.18%
Ratio	0.45	–0.38	0.44	0.85	0.94	0.96	0.99	0.95
Skewness	–0.37	–0.21	0.09	0.71	0.04	–0.36	–0.01	–0.39
Kurtosis	0.25	0.94	–0.39	0.99	0.28	1.14	1.14	0.83
Hit Rate	0.56	0.56	0.46	0.56	0.67	0.64	0.64	0.62

EXHIBIT 9

Portfolio Performance for Different Strategies (test data)

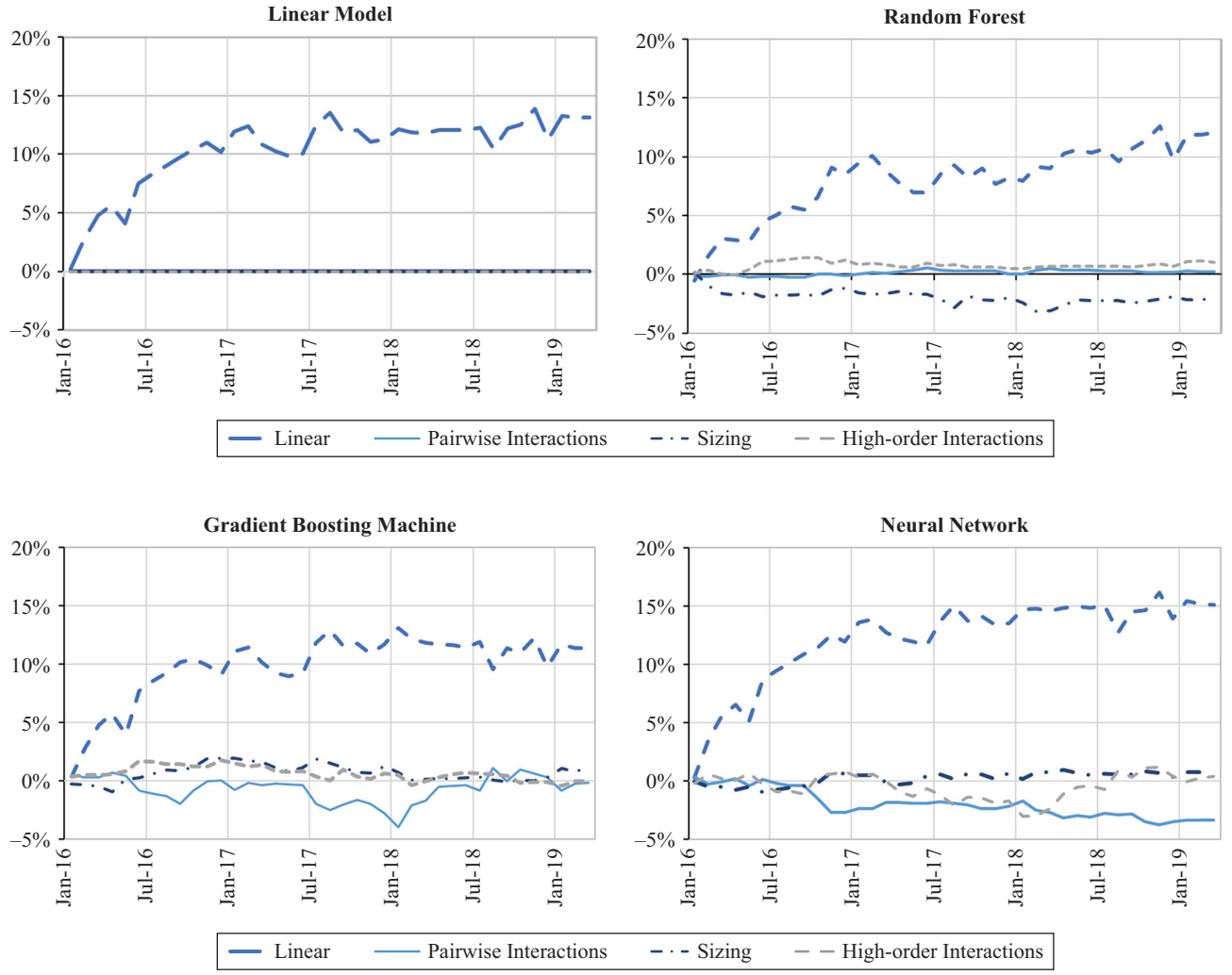


CONCLUSION

We argue that developing reliable and intuitive interpretation is essential for the application of machine learning to investing. We introduce a framework for computing the fingerprint of machine learning models to summarize the linear, nonlinear, and pairwise and high-order interaction effects that drive both predictions and performance. The framework we propose is general and applicable to any predictive model, including base and ensemble models. We find that the results are highly intuitive and informative in a real-world application to currency prediction. Interestingly, our findings reveal as much about the similarities between models as they do about the differences. Despite possible concerns about complex models and their propensity to overfit, we find that random forest, gradient boosting machine, and neural network predicted linear effects that are nearly indistinguishable from those of an ordinary linear regression. Where the models do behave differently, the fingerprint decompositions help identify the most important components of prediction and performance,

EXHIBIT 10

Subcomponent Cumulative Performances (test data)



allowing for further analysis and deeper understanding. We believe that machine learning holds great promise for financial prediction and insight, but it is crucial for investors to apply domain expertise and intuition as part of the process. Additional tools will be required to meet this need. Most importantly, we argue that it is not necessary to view machine learning as a black box. People often trust and learn from others' perspectives despite an incomplete understanding of how that person's brain or thought process actually works. Likewise, we can derive insights from machine learning models if we understand their tendencies and personalities, or more aptly, their *machinalities*.

REFERENCES

Breiman, L. 2001. "Statistical Modeling: The Two Cultures." *Statistical Science* 16 (3): 199–215.

Burnside, C., M. Eichenbaum, and S. Rebelo. 2011. "Carry Trade and Momentum in Currency Markets." *Annual Review of Financial Economics* 3 (1): 511–535.

Czasonis, M., B. Pamir, and D. Turkington. 2019. "Carry On." *The Journal of Alternative Investments* 22 (2): 100–111.

Friedman, J. H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics* 29 (5): 1189–1232.

Friedman, J. H., and B. E. Popescu. 2008. "Predictive Learning via Rule Ensembles." *The Annals of Applied Statistics* 2 (3): 916–954.

Goodfellow, I., Y. Bengio, and A. Courville. *Deep Learning.* Cambridge, MA: MIT Press, 2016.

Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy. 2018. "A Simple and Effective Model-Based Variable Importance Measure." arXiv preprint, arXiv:1805.04755.

Gu, S., B. T. Kelly, and D. Xiu. "Empirical Asset Pricing via Machine Learning." Chicago Booth Research Paper No. 18-04, 2019.

Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2008.

Heaton, J. B., N. G. Polson, and J. H. Witte. 2016. "Deep Learning for Finance: Deep Portfolios." *Applied Stochastic Models in Business and Industry* 33 (1): 3–12.

Kritzman, M., and Y. Li. 2010. "Skulls, Financial Turbulence, and Risk Management." *Financial Analysts Journal* 66 (5): 30–41.

Kuhn, M. 2008. "Building Predictive Models in R Using the Caret Package." *Journal of Statistical Software* 28 (5), https://www.jstatsoft.org/article/view/v028i05.

Kuhn, M., and K. Johnson. *Applied Predictive Modeling.* New York: Springer, 2013.

López de Prado, M. 2019. "Beyond Econometrics: A Roadmap Towards Financial Machine Learning." Working paper, 2019.

Molnar, C. 2019. "Interpretable Machine Learning; A Guide for Making Black Box Models Explainable." https://christophm.github.io/interpretable-ml-book.

Rasekhschaffe, K. C., and R. C. Jones. 2019. "Machine Learning for Stock Selection." *Financial Analyst Journal* 75 (3): 70–88.

Simonian, J., and F. J. Fabozzi. 2019. "Triumph of the Empiricists: The Birth of Financial Data Science." *The Journal of Financial Data Science* 1 (1): 12–18.

Turkington, D., and A. Yazdani. 2019. "The Equity Differential Factor in Currency Markets." Working paper.

ADDITIONAL READING

Carry On

MEGAN CZASONIS, BAYKAN PAMIR, AND DAVID TURKINGTON
The Journal of Alternative Investments
https://jai.pm-research.com/content/22/2/100

ABSTRACT: *The carry trade in foreign currencies is known for delivering positive returns, on average, and for occasionally suffering large losses. While these characteristics prevail, on average, across time and across currency pairs, the authors find that interest rate differentials on their own are not sufficient to identify conditions in which currencies reliably exhibit these return and risk attributes. They use three variables—valuation, crowding, and volatility—to identify time periods and cross-sections of currencies in which the carry trade performs best. They document a substantial difference in performance between the carry trade applied to high-volatility versus low-volatility currency pairs. In the full sample from 1984 to 2017, carry in high-volatility pairs has consisted of currencies that are undervalued, on average, experience greater swings in valuation, and have boom and bust cycles aligned with investor crowding. This finding is consistent with the notion that carry represents a risk premium. Carry in low-volatility pairs has the opposite characteristics. Though both strategies performed well prior to the 2008 financial crisis, only carry in high-volatility pairs has worked since.*

Triumph of the Empiricists: *The Birth of Financial Data Science*

JOSEPH SIMONIAN AND FRANK J. FABOZZI
The Journal of Financial Data Science
https://jfds.pm-research.com/content/1/1/10

ABSTRACT: *The authors situate financial data science within the broader history of econometrics and argue that its ascendance marks a reorientation of the field toward a more empirical and pragmatic stance. They also argue that owing to the unique nature of financial information, financial data science should be considered a field in its own right and not just an application of data science methods to finance.*