

# Forecasting currency covariances using machine learning tree-based algorithms with low and high prices<sup>1</sup>

Sylwester Bejger,<sup>a</sup> Piotr Fiszeder<sup>b</sup>

**Abstract.** We combine machine learning tree-based algorithms with the usage of low and high prices and suggest a new approach to forecasting currency covariances. We apply three algorithms: Random Forest Regression, Gradient Boosting Regression Trees and Extreme Gradient Boosting with a tree learner. We conduct an empirical evaluation of this procedure on the three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are predominantly more accurate than the Dynamic Conditional Correlation model based on closing prices. The results of the analyses indicate that the GBRT algorithm is the best-performing method.

**Keywords:** machine learning, tree-based ensembles, volatility models, high-low range, covariance forecasting

**JEL:** C22, C45, C53, C58, C63, G17

## 1. Introduction

Multivariate volatility models can be used in many financial applications, such as asset pricing, portfolio optimisation, risk management, the estimation of systemic risk in banking, Value-at-Risk estimation or asset allocation. Volatility models of financial instruments that are commonly used are largely based on closing prices only. However, the use of daily low and high prices leads to more accurate estimates and forecasts of variances (e.g. Chou, 2005; Fiszeder & Perczak, 2016; Lin et al., 2012; Molnár, 2016) and covariances (e.g. Chou et al., 2009; Fiszeder, 2018; Fiszeder et al., 2019; Fiszeder & Fałdziński, 2019). Daily low and high prices are almost always available alongside closing prices in financial series. Therefore, making use of them in volatility models is important from a practical viewpoint. The review of multivariate range-based models can be found in Petropoulos et al. (in press).

Recently, the importance of machine learning (ML) algorithms in the forecasting of financial time series has increased considerably (see e.g. de Prado, 2018). ML models, unlike classic (nonlinear) time series analysis, do not require prior

---

<sup>1</sup> This research was financed by the following National Science Centre projects: 2016/21/B/HS4/00662 entitled 'Multivariate volatility models – the application of low and high prices' and 2019/35/B/HS4/00642 entitled 'Forecasting financial markets using a machine learning approach'.

<sup>a</sup> Nicolaus Copernicus University in Toruń, Faculty of Economic Sciences and Management, Department of Applied Informatics and Mathematics in Economics, ul. Gagarina 13a, 87-100 Toruń, Poland, e-mail: sylwester.bejger@umk.pl, ORCID: <https://orcid.org/0000-0001-7900-946X>.

<sup>b</sup> Nicolaus Copernicus University in Toruń, Faculty of Economic Sciences and Management, Department of Econometrics and Statistics, ul. Gagarina 13a, 87-100 Toruń, Poland, e-mail: piotr.fiszeder@umk.pl, ORCID: <https://orcid.org/0000-0002-9777-2239>.

assumptions about the underlying structure of data (Zhang, 2003) and are able to capture recurring nonlinear patterns in time series (see e.g. Fischer et al., 2019). These factors cause machine learning algorithms to outperform most traditional stochastic methods in financial market forecasting (Fiszeder & Orzeszko, 2021; Ryll & Seidens, 2019). The most popular ML approaches in the field of finance are Artificial Neural Network (ANN) and Support Vector Machine (SVM). On average, recurrent neural networks outperform feed-forward neural networks as well as support vector machines (Ryll & Seidens, 2019). However, most models based on ANN and SVM are treated as ‘black box’ algorithms. A black box model is a system that does not reveal its internal mechanisms, and therefore in machine learning it describes models that cannot be understood by looking at their parameters (e.g. an artificial neural network). Interpretable or Explainable Machine Learning refers to methods and models that make the behaviour and predictions of machine learning systems understandable to humans, which is essential in business forecasting and decision-making (Bejger & Elster, 2020). As an alternative to ANN and SVM models, we want to evaluate the performance of tree-based ensemble algorithms in the forecasting of financial time series. Although ensemble learning algorithms are also referred to as black boxes, if a part of an ensemble is a decision tree, the interpretability of the model and predictions becomes much greater. The decision mechanism (model) of a single decision tree is entirely transparent and interpretable (Barredo Arrieta et al., 2019) due to its intrinsic properties. Among others things, it enables the ranking of the relative significance of predictor variables through variable importance metrics (VIMs) (Biau & Scornet, 2016; Breiman, 2001). In an ensemble of trees, these measures could be used for the global and local interpretability of predictions.

The aim of the paper is to suggest a new approach to forecasting currency covariances based on the combination of machine learning tree-based algorithms with the use of low and high prices. The methods we selected are based on the regression tree concept and Classification and Regression Trees (CART) split criterion (Breiman et al., 1984). We apply the Random Forest Regression (RFR) algorithm (Breiman, 2001), the Gradient Boosting Regression Trees (GBRT) algorithm (Friedman, 2001), and the Extreme Gradient Boosting with tree learner (XGBoost, described in Chen & Guestrin, 2016). Although the most popular ML approaches in the field of finance are ANN and SVM learning methods (Henrique et al., 2019; Ryll & Seidens, 2019), we decided to apply the three above-mentioned algorithms instead, for the following reasons:

- models based on ANN and SVM are treated as ‘black box’ algorithms with no clear interpretation of hyperparameters and the importance of features, while tree-based methods are interpretable through VIMs (an intrinsic property of tree-based models) and a well-defined hyperparameter meaning;

- they can handle heterogeneous data (ordered or categorical variables, or a mix of both) with minimal preprocessing;
- they can handle highly correlated predictor variables;
- as all tree-based methods, they intrinsically implement feature selection;
- they either do not overfit (RFR) or are easy to control against overfitting (GBRT, XGBoost);
- tree-based methods are still rarely used in forecasting financial markets, despite the fact that they proved to be among the best in competitions such as M4, M5, or those organised by the Kaggle portal.

The applications of the RFR, GBRT, and XGBoost algorithms in the forecasting of financial markets are presented in the works of Ghosh et al. (in press), Islam et al. (2021), Khaidem et al. (2016), Krauss et al. (2017), Kumar and Thenmozhi (2006), Waldow et al. (2021), Yang (2021) and Yang et al. (2021). However, most of those studies are devoted to the forecasting of stock prices or exchange rates. To the best of our knowledge, this study presents the first application of the forecasting of currency covariances. We empirically evaluate the usability of the algorithms on the three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY, and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are predominantly more accurate than the Dynamic Conditional Correlation benchmark model based on closing prices.

The remaining part of the paper is organised in the following way: Section 2 describes applied models and methods, in Section 3 we present the data and an outline of the study, Section 4 evaluates the forecasts of the covariance of returns from the RFR, GBRT, XGBoost algorithms and the DCC model, and Section 5 contains the conclusions of the study.

## 2. Theoretical background

### 2.1. Tree-based ensemble algorithms

The building block of the machine learning algorithms which we use is a weak learner of a regression tree (e.g. Breiman et al., 1984; Quinlan, 1992). A regression tree is a supervised learning method used to learn a function that combines a set of variables intending to predict another variable. The general idea of a tree learner is to partition feature space  $X$  into a set of rectangles and then fit a simple model (like a constant  $c$ ) in each one (Hastie et al., 2009).

The prediction function of a tree is defined as:

$$f(x) = \sum_{m=1}^M c_m I(x, R_m), \quad (1)$$

where  $M$  is the number of leaves in the tree;  $R_m$  is a region in the feature space (corresponding to leaf  $m$ ),  $c_m$  is a constant corresponding to region  $m$ ,  $I$  is the indicator function (returning 1 if  $x \in R_m$ , 0 otherwise). The value of  $c_m$  is determined in the training phase of the tree. For regression, we partition the predictor space to find a set of regions  $R$  that minimise the RSS, given by:

$$\sum_{m=1}^M \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2, \quad (2)$$

where  $\hat{y}_{R_m}$  is the mean response for the training observations within the  $m$ -th region.

As it is infeasible to consider every possible partition of the feature space into  $M$  regions, a top-down, greedy algorithm known as recursive partitioning (e.g., CART for a binary tree, Breiman et al., 1984) is used to train the single tree. The essential element of CART is a split criterion, dependent on an impurity measure. A regression tree split criterion computes the (renormalised) difference between the empirical variance in the node before and after a cut is performed.

### 2.1.1. Random Forest ensemble algorithm

A single regression tree is typically a weak prediction model which is unstable (high variance learner). To reduce prediction variance and prevent bias from increasing, one can combine the prediction of many weak learners (Schapire, 1990), creating an ensemble of learners. A Random Forest (Breiman, 2001; Ho, 1998) is an ensemble (or forest) of decision trees grown from a randomised variant of a tree induction algorithm.

The Random Forest exploits two sources of randomness to reduce the correlation of residuals of base learners, which decreases the general prediction error. The first of them is a bootstrap, where each tree is constructed on the basis of a bootstrap-resampled training data set, thanks to which the trees are different from each other. The second is a split-variable randomisation: each time a split is to be performed, the search for the split variable is limited to a random  $m$  subset of the  $p$  predictors, which leads to the decorrelation of trees. When the forest's element is a regression tree, such a learning mechanism is called Random Forest Regression.

The RFR prediction is the unweighted average over predictions (1) of the set of  $K$  trees:

$$F(x) = \frac{1}{K} \sum_j^K f_k(x), \quad (3)$$

If  $m = p$ , we have an ancestor of random forests, bootstrap aggregation, or a bagging ensemble method (Breiman, 2004).

### 2.1.2. Gradient Boosting Regression Trees

Boosting (Schapire, 1999) is also a technique which additively combines many weak learners to an ensemble. It is a form of a more general concept of additive expansion learning. Boosting algorithms evolved from Adaboost, the first successful boosting algorithm (Freund, 1995; Freund & Schapire, 1997), to its generalisation as a Gradient Boosting that handles various loss functions (Friedman, 2001; Friedman et al., 2000). The GBRT algorithm involves two main steps – fitting (shallow) decision tree  $h$  to the ‘residuals’ from the model, given current tree model  $F$ , and updating  $F$  by adding  $h$  and updating the residuals. Those steps are repeated until an error on the test set starts to arise. The natural idea is to generalise boosting for any differentiable loss function (for example, not sensitive to outliers). In our study, we use the Huber loss function of the following form:

$$L(y, F) = \begin{cases} \frac{1}{2}(y - F)^2 & |y - F| \leq \delta, \\ \delta \left( |y - F| - \frac{\delta}{2} \right) & |y - F| > \delta. \end{cases} \quad (4)$$

The most important differences between GBRT and RFR are the following: trees are grown sequentially, which means that each tree is grown using information from the previously grown trees; the method is more sensitive to overfitting, and the number of trees should be controlled ex-post.

### 2.1.3. Extreme Gradient Boosting algorithm

XGBoost is a scalable machine learning system for tree boosting. It was implemented and described by Chen and Guestrin (2016). XGBoost is widely recognised by practitioners (e.g. Kaggle competitors) and has implementations in many programming languages (R, Python, Java, Scala, Julia, Perl, and others). The method is based on the GBRT idea, but the computational implementation offers more hyperparameters to tune. There is a technical difference in optimising a loss function between GBRT and XGBoost, as GBRT divides the optimisation problem into two parts (the determination of the direction of the minimisation step, the optimisation of the step length). XGBoost tries to determine both in one step directly. It means that at each iteration, both algorithms need to calculate the gradient at the current estimate. Still, XGBoost also needs to calculate the Hessian matrix, so the XGBoost loss function must be twice differentiable.

## 2.2. Range-based covariance estimator for exchange rates

In the suggested approach to covariance forecasting, we apply the estimator of the covariance of returns calculated on the basis of low and high prices. This estimator has an advantage over that based only on the closing prices, because it uses information about the price changes during the day. Let us consider two exchange rates of currencies  $A$  and  $B$  in terms of currency  $C$ , denoted as  $A/C$  and  $B/C$ , respectively. In the absence of triangular arbitrage opportunities, the return of the cross rate can be written as:

$$\Delta \ln A/B = \Delta \ln A/C - \Delta \ln B/C. \quad (5)$$

Then, the range-based estimator of covariance for the currency pairs can be represented as:

$$\text{cov}(\Delta \ln A/C, \Delta \ln B/C) = 0.5[\text{var}(\Delta \ln A/C) + \text{var}(\Delta \ln B/C) - \text{var}(\Delta \ln A/B)]. \quad (6)$$

As a variance estimator, we use the Parkinson (1980) estimator given as:

$$\text{var}_{Pt} = [\ln(H_t/L_t)]^2 / (4 \ln 2), \quad (7)$$

where  $H_t$  and  $L_t$  are the daily high and low prices, respectively.

More details about the applied range-based covariance estimator and its properties can be found in Fiszeder and Orzeszko (2021), who employ this estimator in a new methodology for dynamic modeling and forecasting covariance matrices based on support vector regression.

## 2.3. The DCC model

In this section, we describe the DCC model of Engle (2002). It is one of the most popular multivariate volatility models (see e.g. Bauwens et al., 2012) and is often used as a benchmark model in empirical studies. Let us assume that  $\boldsymbol{\varepsilon}_t$  ( $N \times 1$  vector) is the innovation process for the conditional mean and can be written as:

$$\boldsymbol{\varepsilon}_t | \boldsymbol{\psi}_{t-1} \sim N(0, \mathbf{cov}_t), \quad (8)$$

where  $\boldsymbol{\psi}_{t-1}$  is the set of all information available at time  $t - 1$ ,  $N$  is the multivariate normal distribution, and  $\mathbf{cov}_t$  is the  $N \times N$  symmetric conditional covariance matrix.

The DCC( $P, Q$ ) model can be presented as:

$$\mathbf{cov}_t = \mathbf{D}_t \mathbf{cor}_t \mathbf{D}_t, \quad (9)$$

$$\mathbf{cor}_t = \mathbf{Q}_t^{*-1} \mathbf{Q}_t \mathbf{Q}_t^{*-1}, \quad (10)$$

$$\mathbf{Q}_t = \left(1 - \sum_{i=1}^Q \zeta_i - \sum_{j=1}^P \theta_j\right) \mathbf{S} + \sum_{i=1}^Q \zeta_i (\mathbf{z}_{t-i} \mathbf{z}'_{t-i}) + \sum_{j=1}^P \theta_j \mathbf{Q}_{t-j}, \quad (11)$$

where  $\mathbf{D}_t = \text{diag}(h_{1t}^{1/2}, h_{2t}^{1/2}, \dots, h_{Nt}^{1/2})$ , conditional variances  $h_{kt}$  (for  $k = 1, 2, \dots, N$ ) are described as univariate GARCH models (equations (12–13)),  $\mathbf{z}_t$  is the standardised  $N \times 1$  residual vector assumed to be serially independently distributed given as  $\mathbf{z}_t = \mathbf{D}_t^{-1} \boldsymbol{\varepsilon}_t$ ,  $\mathbf{cor}_t$  is the time varying  $N \times N$  conditional correlation matrix of  $\mathbf{z}_t$ ,  $\mathbf{S}$  is the unconditional  $N \times N$  covariance matrix of  $\mathbf{z}_t$  (it can also be estimated with other parameters of the model, but this makes estimation more difficult) and  $\mathbf{Q}_t^*$  is the diagonal  $N \times N$  matrix composed of the square root of diagonal elements of  $\mathbf{Q}_t$ . Parameters  $\zeta_i$  (for  $i = 1, 2, \dots, Q$  and  $\theta_j$  (for  $j = 1, 2, \dots, P$ ) are nonnegative and satisfy the  $\sum_{i=1}^Q \zeta_i + \sum_{j=1}^P \theta_j < 1$  condition.

The univariate GARCH( $p, q$ ) model applied in the DCC model can be written as:

$$\varepsilon_{kt} | \psi_{t-1} \sim N(0, h_{kt}), \quad k = 1, 2, \dots, N, \quad (12)$$

$$h_{kt} = \alpha_{k0} + \sum_{i=1}^q \alpha_{ki} \varepsilon_{k,t-i}^2 + \sum_{j=1}^p \beta_{kj} h_{k,t-j}, \quad (13)$$

where  $\alpha_{k0} > 0, \alpha_{ki} \geq 0, \beta_{kj} \geq 0$  (for  $k = 1, 2, \dots, N; i = 1, 2, \dots, q; j = 1, 2, \dots, p$ ), weaker conditions for nonnegativity of the conditional variance can be assumed (see Nelson & Cao, 1992). The requirement for covariance stationarity of  $\varepsilon_{kt}$  is  $\sum_{i=1}^q \alpha_{ki} + \sum_{j=1}^p \beta_{kj} < 1$ .

Parameters of the DCC model can be estimated by the quasi-maximum likelihood method using a two-stage approach. Let the parameters of model  $\boldsymbol{\Theta}$  be written in two groups, i.e.  $\boldsymbol{\Theta}' = (\boldsymbol{\Theta}'_1, \boldsymbol{\Theta}'_2)$ , where  $\boldsymbol{\Theta}_1$  is the vector of the parameters of conditional means and variances, and  $\boldsymbol{\Theta}_2$  is the vector of the parameters of the correlation part of the model. The log-likelihood function can be written as the sum of two parts:

$$L(\boldsymbol{\Theta}) = L_{Vol}(\boldsymbol{\Theta}_1) + L_{Corr}(\boldsymbol{\Theta}_2 | \boldsymbol{\Theta}_1), \quad (14)$$

where  $L_{Vol}(\boldsymbol{\Theta}_1)$  represents the volatility part:

$$L_{Vol}(\boldsymbol{\Theta}_1) = -\frac{1}{2} \sum_{t=1}^n (N \ln(2\pi) + \ln |\mathbf{D}_t|^2 + \boldsymbol{\varepsilon}'_t \mathbf{D}_t^{-2} \boldsymbol{\varepsilon}_t), \quad (15)$$

while  $L_{Corr}(\Theta_2|\Theta_1)$  can be viewed as the correlation component:

$$L_{Corr}(\Theta_2|\Theta_1) = -\frac{1}{2}\sum_{t=1}^n(\ln|\mathbf{cor}_t| + \mathbf{z}'_t\mathbf{cor}_t^{-1}\mathbf{z}_t - \mathbf{z}'_t\mathbf{z}_t). \quad (16)$$

$L_{Vol}(\Theta_1)$  can be written as the sum of the log-likelihood functions of  $N$  univariate GARCH models:

$$L_{Vol}(\Theta_1) = -\frac{1}{2}\sum_{k=1}^N\left(n\ln(2\pi) + \sum_{t=1}^n\left(\ln(h_{kt}) + \frac{\epsilon_{kt}^2}{h_{kt}}\right)\right). \quad (17)$$

In the first stage, the parameters of univariate GARCH models can be estimated separately for each of the assets and the estimates of  $h_{kt}$  can be obtained. In the second stage, residuals transformed by their estimated standard deviations are used to estimate the parameters of the correlation part ( $\Theta_2$ ) conditioning on the parameters estimated in the first stage ( $\hat{\Theta}_1$ ).

### 3. Data and description of the research

We evaluate the accuracy of the proposed procedure of covariance forecasting based on data from the Forex market, for the purpose of which we examine three most heavily traded currency pairs, namely EUR/USD, USD/JPY and GBP/USD. Daily data for the period from 2 January 2004 to 30 December 2016 are used. A total sample of 3,365 observations is split into a training set (period: 14 January 2004 to 31 December 2014, size: 2,846 observations, the first eight observations are truncated during the construction of the analytical dataset) and a test set (period: 2 January 2015 to 30 December 2016, holdout set size: 519 observations).

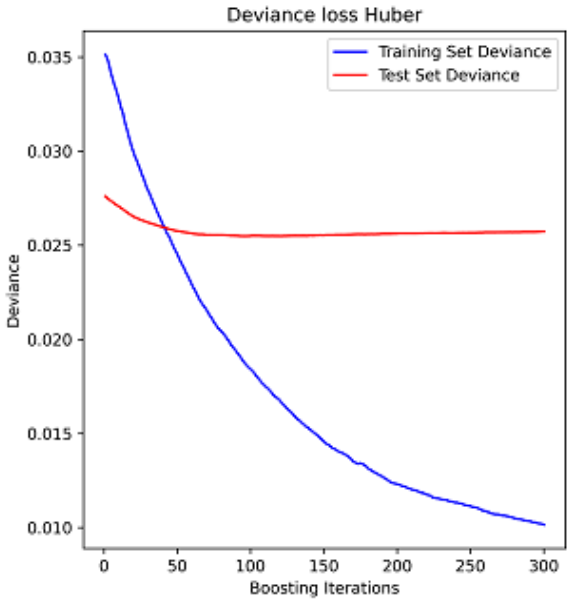
The target variable is the covariance of returns of currency pairs given in (6). This estimator is more efficient than the one based on closing prices only. The set of predictors contains the following time series: – target\_lag\_1 until target\_lag\_8, min\_A/C\_lag1, min\_B/C\_lag1, max\_A/C\_lag1, max\_B/C\_lag1, close\_A/C\_lag1, close\_B/C\_lag1, lnzwr\_A/C\_lag1 and lnzwr\_B/C\_lag1. We also add categorical predictors commonly used in training time series models: month, weekofyear, dayofweek, dayofyear, dayofmonth. Categorical variables are integer-coded (which is a better option for the tree-based methods than one-hot encoding).

We implement machine learning pipelines for the random forest, GBRT and XGBoost in the Python 3.6.3 environment, and use standard libraries for data processing and machine learning, i.e. numpy, scikit-learn, pandas and datetime. Additionally, we use the XGBoost library and the scikit-learn wrapper interface for XGBoost. The training set, containing about 85% of the samples, is used to tune



hyperparameters using the time series k-fold cross-validator (*TimeSeriesSplit* method). Standard cross-validation techniques assume that samples are independent and identically distributed, and would result in an unreasonable correlation between training and testing instances (yielding poor estimates of the generalisation error) on time series data. In the *TimeSeriesSplit* method, successive training sets are supersets of those that come before them. It also adds all surplus data to the first training partition, which is always used to train the model. The preliminary tuning of hyperparameters is done by searching the space of the parameters (*RandomizedSearchCV* or *GridSearchCV* methods) with the above described k-fold cross-validation. In a random forest, the additional calibration of the *min\_samples\_split*, *max\_depth*, *min\_impurity\_decrease* and *min\_samples\_leaf* split hyperparameters is performed. In the case of boosting (GBRT), the ex-post control for overfitting (the value of the *n\_estimators* hyperparameter) is done (see the Figure).

**Figure.** Train and test set deviance against boosting iterations



Source: authors' calculations.

For tree-based supervised learning, the critical model's elements are impurity measures, which determine the split quality and a loss function, influencing the quality of predictions. In regression on time series data, the MSE and RMSE impurity measures are applied. As a loss function, we utilise the Huber loss function in the GBRT model, and the square loss in XGBoost.

The three ensemble models are tuned, trained, and applied on the out-of-sample instances to generate forecasts. Parameters of the DCC model are estimated each day on a rolling sample of a fixed size of 500 observations (approximately 2 years).

#### 4. Comparison of covariance forecasts

This section compares out-of-sample one-day-ahead forecasts of the covariance of returns from three machine learning algorithms (RFR, GBRT and XGBoost), with the forecasts from the DCC model. We evaluate forecasts for a two-year period from 2 January 2015 to 30 December 2016.

The sum of products of 15-minute returns (the realised covariance) is employed as a proxy of the daily covariance for the evaluation of the forecasts. We assess the forecasts from the models based on two primary measures, i.e. the mean squared error (MSE) and the mean absolute error (MAE). In order to evaluate the statistical significance of the results, the Diebold-Mariano test (Diebold & Mariano, 1995) is applied. We perform a pairwise comparison with respect to the DCC benchmark model. The forecasting performance results are presented in the Table.

**Table.** Evaluation of covariance forecasts for selected exchange rates

Method	Forecast evaluation criteria			
	MSE	DM test <i>p</i> -value	MAE	DM test <i>p</i> -value
EUR/USD-JPY/USD				
DCC .....	0.2016	–	0.1741	–
RFR .....	0.1935	0.0420	<b>0.1455</b>	0.0000
GBRT .....	<b>0.1889</b>	0.0004	0.1477	0.0000
XGBoost .....	0.1925	0.0445	0.1464	0.0000
EUR/USD-GBP/USD				
DCC .....	0.3122	–	0.1662	–
RFR .....	0.2905	0.1416	0.1870	0.0130
GBRT .....	<b>0.2799</b>	0.0258	<b>0.1536</b>	0.1060
XGBoost .....	0.2903	0.0729	0.1833	0.0199
JPY/USD-GBP/USD				
DCC .....	0.7503	–	0.1717	–
RFR .....	<b>0.6408</b>	0.0132	0.1407	0.0085
GBRT .....	0.6476	0.0141	<b>0.1373</b>	0.0053
XGBoost .....	0.6349	0.0158	0.1486	0.0545

Note: the evaluation period is from 2 January 2015 to 30 December 2016, the realised covariances are used as the real values of covariance and estimated as the sum of products of 15-min. returns. The lowest values of the MSE and MAE are marked in bold. The *p*-values of the Diebold-Mariano test are presented for pairs of models: the selected algorithm and the DCC benchmark. A *p*-value lower than the significance level means that the forecasts of covariance from the selected method with a lower evaluation measure are significantly more accurate.

Source: authors' calculations.

Under the MSE criterion, the forecasts of covariance from all the three machine learning algorithms are more accurate than the forecasts based on the DCC model. According to the Diebold-Mariano test, the advantage of these algorithms is statistically significant at the level of 10%, except the EUR/USD-GBP/USD relation for RFR. Under the MAE measure, the forecasts based on the analysed machine learning methods are again significantly more accurate than the forecasts from the DCC model for the EUR/USD-JPY/USD and JPY/USD-GBP/USD relations. For EUR/USD-GBP/USD, the lowest value of the criterion occurs for the GBRT algorithm, but this result is not statistically significant. Predominantly, both of the loss functions indicate the GBRT algorithm as the best performing method.

## 5. Conclusions

The machine learning ensemble method is a method that combines a set of weak learners to create a (more potent) learner that performs better than any of the individual ones. Ensemble methods help reduce bias and/or variance. We use a decision tree as a base, weak learner. We examine the performance of three popular tree-based ensemble algorithms: random forest (regression), GBRT and XGBoost. These algorithms exploit two different approaches to ensemble learning. Random forest trains individual estimators independently over bootstrapped subsets of data (bagging) and incorporates the second level of randomness. When optimising each node split, only a random subsample (without replacement) of the attributes will be evaluated, with the purpose of the further decorrelating of the estimators. Both GBRT and XGBoost utilise a boosting technique that is different from a random forest. In boosting, individual trees are fitted sequentially, observations are weighted differently in each iteration, and poor-performing trees are excluded. All the three algorithms belong to a group of machine learning algorithms which are most popular and widely-used in many fields. It is worth noticing that a variant of the gradient boosting algorithm, LightGBM, has won the M5 forecasting competition (Makridakis et al., 2020). Tree-based ensembles are also becoming increasingly popular in financial forecasting (Henrique et al., 2019; de Prado, 2018).

Daily low and high prices contain important information about the variability of the prices of financial instruments, but they are very seldom used for the estimation of volatility models. We combine machine learning tree-based algorithms with the usage of low and high prices and suggest a new approach to forecasting currency covariances. We conduct an empirical evaluation of this procedure on the basis of three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY

and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are in most part more accurate than the DCC model, used as a benchmark model based on closing prices. The results of the analyses indicate that the GBRT algorithm is the best-performing method.

Research on tree-based machine learning methods in covariance forecasting can be further developed, for example in the area of the analysis of the importance of predictors or studies on the interpretability of the optimal values of hyper-parameters. Other issues, such as modifying a loss function in boosting-based methods and examining the performance of random forest and gradient boosting variants (e.g. quantile regression forests, dynamic random forests, and the LightGBM algorithm) seem worth further investigation as well.

## References

- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2019, December 26). *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. <https://arxiv.org/pdf/1910.10045.pdf>.
- Bauwens, L., Hafner, C., & Laurent, S. (Eds.). (2012). *Handbook of Volatility Models and Their Applications*. John Wiley & Sons. <https://doi.org/10.1002/9781118272039>.
- Bejger, S., & Elster, S. (2020). Artificial Intelligence in economic decision making: how to assure a trust?. *Ekonomia i Prawo / Economics and Law*, 19(3), 411–434. <https://doi.org/10.12775/EiP.2020.028>.
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, 25(2), 197–227. <https://doi.org/10.1007/s11749-016-0481-7>.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Breiman, L. (2004). Bagging predictors. *Machine Learning*, 24, 123–140. <https://doi.org/10.1007/BF00058655>.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees* (1st edition). Chapman & Hall/CRC. <https://doi.org/10.1201/9781315139470>.
- Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco.
- Chou, R. Y. (2005). Forecasting Financial Volatilities with Extreme Values: The Conditional Autoregressive Range (CARR) Model. *Journal of Money, Credit and Banking*, 37(3), 561–582. <https://doi.org/10.1353/mcb.2005.0027>.
- Chou, R. Y., Wu, C. C., & Liu, N. (2009). Forecasting time-varying covariance with a range-based dynamic conditional correlation model. *Review of Quantitative Finance and Accounting*, 33(4), 327–345. <https://doi.org/10.1007/s11156-009-0113-3>.

- Diebold, F. X., & Mariano, R. S. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263. <https://doi.org/10.2307/1392185>.
- Engle, R. F. (2002). Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models. *Journal of Business and Economic Statistics*, 20(3), 339–350. <https://doi.org/10.1198/073500102288618487>.
- Fischer, T. G., Krauss, C., & Deinert, A. (2019). Statistical Arbitrage in Cryptocurrency Markets. *Journal of Risk and Financial Management*, 12(1), 1–15. <https://doi.org/10.3390/jrfm12010031>.
- Fiszeder, P. (2018). Low and high prices can improve covariance forecasts: The evidence based on currency rates. *Journal of Forecasting*, 37(6), 641–649. <https://doi.org/10.1002/for.2525>.
- Fiszeder, P., & Faldziński, M. (2019). Improving forecasts with the co-range dynamic conditional correlation model. *Journal of Economic Dynamics and Control*, 108, 1–16. <https://doi.org/10.1016/j.jedc.2019.103736>.
- Fiszeder, P., Faldziński, M., & Molnár, P. (2019). Range-based DCC models for covariance and value-at-risk forecasting. *Journal of Empirical Finance*, 54, 58–76. <https://doi.org/10.1016/j.jempfin.2019.08.004>.
- Fiszeder, P., & Orzeszko, W. (2021). Covariance matrix forecasting using support vector regression. *Applied Intelligence*, 51(10), 7029–7042. <https://doi.org/10.1007/s10489-021-02217-5>.
- Fiszeder, P., & Perczak, G. (2016). Low and high prices can improve volatility forecasts during periods of turmoil. *International Journal of Forecasting*, 32(2), 398–410. <https://doi.org/10.1016/j.ijforecast.2015.07.003>.
- Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, 121(2), 256–285. <https://doi.org/10.1006/inco.1995.1136>.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. <https://doi.org/10.1006/jcss.1997.1504>.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>.
- Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2), 337–407. <https://doi.org/10.1214/aos/1016218223>.
- Ghosh, P., Neufeld, A., & Sahoo, J. K. (in press). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Finance Research Letters*. <https://doi.org/10.1016/j.frl.2021.102280>.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd edition). Springer. <https://doi.org/10.1007/978-0-387-84858-7>.
- Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124, 226–251. <https://doi.org/10.1016/j.eswa.2019.01.012>.
- Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832–844. <https://doi.org/10.1109/34.709601>.

- Islam, S. F. N., Sholahuddin, A., & Abdullah, A. S. (2021). Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah. *Journal of Physics: Conference Series*, 1722, 1–11. <https://doi.org/10.1088/1742-6596/1722/1/012016>.
- Khaidem, L., Saha, S., & Dey, S. R. (2016, April 29). *Predicting the direction of stock market prices using random forest*. <https://arxiv.org/pdf/1605.00003.pdf>.
- Krauss, Ch., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2), 689–702. <https://doi.org/10.1016/j.ejor.2016.10.031>.
- Kumar, M., & Thenmozhi, M. (2006). *Forecasting stock index movement: A comparison of support vector machines and random forest*. 9th Capital Markets Conference, Vashi.
- Lin, E. M. H., Chen, C. W. S., & Gerlach, R. (2012). Forecasting volatility with asymmetric smooth transition dynamic range models. *International Journal of Forecasting*, 28(2), 384–399. <https://doi.org/10.1016/j.ijforecast.2011.09.002>.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020, October 6). *The M5 Accuracy competition: Results, findings and conclusions*. [https://www.researchgate.net/publication/344487258\\_The\\_M5\\_Accuracy\\_competition\\_Results\\_findings\\_and\\_conclusions](https://www.researchgate.net/publication/344487258_The_M5_Accuracy_competition_Results_findings_and_conclusions).
- Molnár, P. (2016). High-low range in GARCH models of stock return volatility. *Applied Economics*, 48(51), 4977–4991. <https://doi.org/10.1080/00036846.2016.1170929>.
- Nelson, D. B., & Cao, C. Q. (1992). Inequality Constraints in the Univariate GARCH Model. *Journal of Business & Economic Statistics*, 10(2), 229–235. <https://doi.org/10.2307/1391681>.
- Parkinson, M. (1980). The Extreme Value Method for Estimating the Variance of the Rate of Return. *The Journal of Business*, 53(1), 61–65. <https://doi.org/10.1086/296071>.
- Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Oliveira, F. L. C., Baets, S. D., Dokumentov, A., Ellison, J., Fiszeder, P., Franses, P. H., Frazier, D. T., Gilliland, M., Gönül, M. S., Goodwin, P., Grossi, L., Grushka-Cockayne, Y., Guidolin, M., Gunter, U., Guo, X., Guseo, R., Harvey, N., Hendry, D. F., Hollyman, R., Januschowski, T., Jeon, J., Jose, V. R. R., Kang, Y., Koehler, A. B., Kolassa, S., Kourentzes, N., Leva, S., Li, F., Litsiou, K., Makridakis, S., Martin, G. M., Martinez, A. B., Meeran, S., Modis, T., Nikolopoulos, K., Önkal, D., Paccagnini, A., Panagiotelis, A., Panapakidis, I., Paviá, J. M., Pedio, M., Pedregal, D. J., Pinson, P., Ramos, P., Rapach, D. E., Reade, J. J., Rostami-Tabar, B., Rubaszek, M., Sermpinis, G., Shang, H. L., Spiliotis, E., Syntetos, A. A., Talagala, P. D., Tashman, L., Thomakos, D., Thorarindottir, T., Todini, E., Arenas, J. R. T., Wang, X., Winkler, R. L., Yusupova, A., & Ziel, F. (in press). *Forecasting: Theory and Practice*. *International Journal of Forecasting*.
- de Prado, M. L. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons.
- Quinlan, J. R. (1992). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.
- Ryll, L., & Seidens, S. (2019, July 6). *Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey*. <https://arxiv.org/pdf/1906.07786.pdf>.

- Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, 5(2), 197–227. <https://doi.org/10.1007/BF00116037>.
- Schapire, R. E. (1999). *A Brief Introduction to Boosting*. Sixteenth International Joint Conference on Artificial Intelligence, Stockholm.
- Waldow, F., Schnaubelt, M., Krauss, C., & Fischer, T. G. (2021). Machine Learning in Futures Markets. *Journal of Risk and Financial Management*, 14(3), 1–144. <https://doi.org/10.3390/jrfm14030119>.
- Yang, Y. (2021). *Market Forecast using XGboost and Hyperparameters Optimized by TPE*. 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou.
- Yang, Y., Wu, Y., Wang, P., & Jiali, X. (2021). *Stock Price Prediction Based on XGBoost and LightGBM*. 2021 International Conference on Economic Innovation and Low-carbon Development, Qingdao.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 59–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0).