Journal of Asset Management (2020) 21:428–446
https://doi.org/10.1057/s41260-020-00167-0

ORIGINAL ARTICLE

Improving CAT bond pricing models via machine learning

Tobias Götze[1] · Marc Gürtler[1] · Eileen Witowski[1]

Revised: 30 April 2020 / Published online: 23 June 2020

Abstract

Enhanced machine learning methods provide an encouraging alternative to forecast asset prices by extending or generalizing the possible model specifications compared to conventional linear regression methods. Even if enhanced methods of machine learning in the literature often lead to better forecasting quality, this is not clear for small asset classes, because in small asset classes enhanced machine learning methods may potentially over-fit the in-sample data. Against this background, we compare the forecasting performance of linear regression models and enhanced machine learning methods in the market for catastrophe (CAT) bonds. We use linear regression with variable selection, penalization methods, random forests and neural networks to forecast CAT bond premia. Among the considered models, random forests exhibit the highest forecasting performance, followed by linear regression models and neural networks.

Keywords CAT bond · Machine learning · Linear regression · Risk premium

JEL Classification C45 · C58 · G12 · G17 · G22

Introduction

Empirical models to forecast the future price of financial assets are predominantly based on linear regression models (Campbell and Thompson 2007; Rapach et al. 2010; Thornton and Valente 2012). A key strength of linear regression models is that the economic relationships between the variables in the model can be understood and interpreted with relatively low effort. Interpretability is important for developing a forecasting model because the modeler can identify the causes for the poor performance of the model relatively easily. However, a model to forecast asset prices must fulfill further requirements and should (1) provide precise estimates over the respective forecasting horizon, (2) be robust toward outliers and (3) build a stable relationship between the dependent and explanatory variables throughout the calibration and forecasting horizon (to be robust toward changing market conditions). In order to fulfill these requirements, an asset pricing model should be based on both statistically and economically significant price determinants and avoid over-fitting issues. Besides, a good forecasting framework relies on the correct specification of the functional relationship between the dependent and explanatory variables and a suitable choice of the underlying conditions of the prediction (Gu et al. 2020). Therefore, the development of an appropriate forecasting model is evidently a complex problem, and linear regression models may not always provide the best solution to that problem.

Thus, enhanced machine learning methods provide a potentially valuable tool to accomplish the above-mentioned modeling challenges by allowing a rich set of possible model specifications compared to conventional methods (Khandani et al. 2010; Mullainathan and Spiess 2017; Gu et al. 2020). While linear regression methods and the types of methods introduced throughout this article all belong to the same model family, we will subsequently use the term "enhanced machine learning method" to describe methods that either extend or generalize linear regression methods. The rich set of model specifications that can be obtained with enhanced machine learning methods also makes these

✉ Marc Gürtler
marc.guertler@tu-bs.de

Tobias Götze
t.goetze@tu-bs.de

Eileen Witowski
e.witowski@tu-bs.de

[1] Department of Finance, Braunschweig Institute of Technology, Abt-Jerusalem-Straße 7, 38106 Brunswick, Germany

methods susceptible to over-fitting, especially when they are applied to relatively small data sets. As over-fitting may result in poor out-of-sample forecasts, a genuine question that arises is whether enhanced machine learning methods can be designed by applying hyperparameter tuning methods to exhibit better forecasting performance than linear regression models. This research question motivates the primary objective of our study: assessing the potential of enhanced machine learning methods in comparison with linear regression models for forecasting asset prices of a small asset class. Therefore, we develop both linear regression- and enhanced machine learning-based forecasting models for risk premia in the market for CAT bonds.

Following the continuous increase in both the frequency and damage intensity of natural catastrophes over the past few decades, the risk for (re-)insurance companies seems to have increased and, in turn, triggered higher prices and capacity constraints for the traditional (re-)insurance market (Gron 1994; Froot 2001; Cummins and Weiss 2009). Therefore, CAT bonds have demonstrated growing utilization as an alternative instrument to conventional reinsurance contracts (Cummins and Weiss 2009). From an investor perspective, CAT bonds are attractive because their returns exhibit a relatively small correlation with other asset classes (Froot 2001; Cummins and Weiss 2009). For the reasons mentioned, pricing of CAT bonds is an increasingly important subject in the literature on insurance and asset pricing.

For two reasons, CAT bonds form an interesting asset class for testing enhanced machine learning methods. First, the CAT bond universe is relatively small, which implies a limited size of training and test data sets and creates a particularly challenging environment for enhanced machine learning methods. Second, scientific literature has already established a good understanding of the major determinants of CAT bond risk premia (Braun 2016; Gürtler et al. 2016). Hence, the in-sample variance of premia over time and across the cross section of CAT bonds can be explained with relatively high precision based on well-known influencing factors. In addition, linear regression models are shown to perform relatively well in the out-of-sample forecast of CAT bond premia (Galeotti et al. 2013; Braun 2016; Trottier et al. 2018) and thus provide a high-performance benchmark for enhanced machine learning methods. The risk of CAT bonds is assessed in an explicit third-party analysis by specialized risk modeling companies that contains a modeled distribution of bond losses and expected loss (EL) as the most characteristic parameter (Lane 2018). The availability of a modeled loss distribution and EL suggests that it should be relatively easy to derive precise price estimates for CAT bonds compared to other asset classes, where risk is represented by a potentially less accurate and more opaque rating (Lane 2018).[1] The availability of risk assessment is among the potential causes for the good performance of linear regression models; consequently, linear regression models are expected to provide a satisfying forecast of CAT bond premia. For the reasons mentioned, it is even more interesting to test whether forecasting models for CAT bond premia can be improved through enhanced machine learning methods.

Our study is based on a data set comprising nearly all CAT bond issues conducted in the time period between 2002 and 2017. Apart from CAT bond data referring to issue volumes, insured peril types and locations, trigger types, maturity terms and sponsors, we incorporate macroeconomic data including the returns of the S&P500, risk premia of corporate bonds, returns on the reinsurance market and a CAT bond price index. We use these data to develop a series of forecasting models for CAT bond premia, which we test in a rolling sample forecast. As the existing empirical literature already provides a good understanding of the determinants of CAT bond premia, our initial selection of potentially relevant variables is based on Braun (2016) and Gürtler et al. (2016).[2] The models introduced subsequently can be distinguished according to the variable selection method. First, we establish a linear regression model and then reduce its complexity by selecting variables based on statistical and economic significance. Second, we adopt variable selection methods, namely forward, backward and stepwise selection. All three methods select variables based on iterative cross-validation and are applied to a linear regression model. Third, we introduce penalization methods (Lasso and Ridge regression methods), which select variables through the introduction of a constraint in the optimization problem of a linear regression model.[3] Finally, we introduce random forests and neural network models. Both methods select model variables implicitly based on the so-called model hyperparameters and the respective optimization algorithms.

In terms of forecasting performance, our results show that the performance of different linear regression models is relatively similar, and the full linear regression model based on Braun (2016) and Gürtler et al. (2016) provides the best forecasting performance in our data set. When we introduce random forests and neural networks, we find that random forests improve the mean (median) forecasting performance

[1] Among others, the studies of Cantor and Packer (1997), Boot et al. (2005), Skreta and Veldkamp (2009) and Bongaerts et al. (2012) show that the rating of a security can be influenced by other factors than only its actual risk.

[2] The focus of our study is the out-of-sample forecast of CAT bond premia. Consequently, we abstain from an in-depth discussion of our in-sample results. For such a discussion, see Braun (2016) and Gürtler et al. (2016).

[3] Precisely, only the Lasso method selects variables from a linear regression model. The Ridge method only shrinks model coefficients toward zero.

in comparison with linear regression in the rolling forecast. Additionally, random forests show a smaller variance in forecasting performance over time than the linear regression models. We believe that this result is especially important for forecasting models because the uncertainty of future market conditions implies high relevance of stable performance over time. In contrast, the neural network's performance depends on the applied test specification and lags both the linear regression models and the random forests.

Our study contributes extensively to the literature on asset pricing and empirical studies on machine learning. This paper is the first to compare different machine learning methods for forecasting CAT bond premia and to provide an approach for tuning hyperparameters in enhanced machine learning methods. Besides these novelties, it can be emphasized that the CAT bond market is much smaller than other asset markets and, therefore, provides a more challenging environment for the application of machine learning methods (Khandani et al. 2010; Gu et al. 2020). Our analysis may help practitioners to assess the potential of machine learning methods for asset pricing in general and, more specifically, in the context of pricing CAT bonds. An important finding we obtain in this context is that machine learning can already perform quite well on a relatively small data set.

The article is structured as follows. "CAT bond pricing and machine learning methods" section provides an overview of the existing CAT bond pricing methods and the enhanced machine learning methods used in our study. "Data" section describes our sample selection, introduces the variables used in the analysis and exhibits descriptive statistics of the data set. "Empirical analysis" section comprises the empirical analysis including an overview of the model comparison procedure, hyperparameter tuning, the results of the out-of-sample forecasts and a graphical model analysis. "Conclusion" section concludes.

CAT bond pricing and machine learning methods

CAT bond pricing methods

The literature has developed a range of different models to forecast the premia of CAT bonds. In one of the first studies, Lane (2000) models the expected excess return from CAT bonds as a log-linear function of the probability of first loss (PFL) and the conditional expected loss (CEL). Wang (2000, 2004) applies a probability transformation to CAT bonds' loss exceedance curve and uses the transformation to predict premia. Galeotti et al. (2013) compare different uni- and multivariate linear regression models to forecast CAT bond premia. In terms of out-of-sample performance, the differences between these models are rather small. Interestingly, the authors find that the inclusion of further explanatory variables, apart from CAT bonds' EL, does not yield large increases in forecasting performance. Furthermore, model comparison indicates that the transformation introduced by Wang (2000, 2004) performs relatively well. Braun (2016) proposes a linear regression model that outperforms existing benchmark models in the forecast of CAT bond premia. Apart from the EL, this model includes indicator variables for multi-territory bonds, bonds that cover US risks, bonds sponsored by Swiss Re, and bonds with an investment grade rating as well as the rate-on-line index of CAT bond returns, and the risk premium of BB-rated corporate bonds. Thus, Braun (2016) suggests that factors, apart from the EL, are relevant for CAT bond pricing. Gürtler et al. (2016) are the first to consider the secondary market for CAT bonds and reach a similar conclusion as Braun (2016) by showing that secondary market premia vary depending on bond-specific and macroeconomic factors. Trottier et al. (2018) forecast premia based on nonlinear utility functions with hyperbolic (constant) absolute risk aversion. Although they thereby provide a theoretically substantiated model of the relationship between risk premium and EL, their model does not outperform existing out-of-sample benchmarks.

Lane (2018) and Makariou et al. (2020) are the first to use the random forests approach to forecast CAT bond premia. Both authors compare a linear regression model to their random forests model, but do not consider other enhanced machine learning models. In addition, Lane (2018) does not perform systematic hyperparameter tuning, which is a central aspect for using enhanced machine learning models to reduce over-fitting problems. While Lane (2018) considers the time structure when splitting his data into training and test data, the analysis of Makariou et al. (2020) is limited to random subsampling. By excluding the time structure of the data, much of the actual information in an out-of-sample forecasting model is lost because the model cannot be tested for its robustness toward time-based shifts in the data set (Braun 2016). Additionally, Makariou et al. (2020) disregard macroeconomic variables in their model, even though the literature shows the relevance of macroeconomic variables in terms of CAT bond pricing (Braun 2016; Gürtler et al. 2016). Our study considers all the mentioned limitations.

An overview of the literature suggests that (linear) regression models are already well-established in the context of pricing CAT bonds. Therefore, the benchmark model that we introduce for our model comparison is based on a linear regression in which the premium $prem_i$ of CAT bond i is modeled using the following equation:

$$prem_i = \alpha + \beta' X_i + u_i, \tag{1}$$

where CAT bond $i = 1, \dots, n$. α is an intercept, $X_i = (x_{i1}, \dots, x_{ip})'$ is a vector of explanatory variables with the coefficient vector $\beta = (\beta_1, \ldots, \beta_p)'$, and u_i is a random error term.

Enhanced machine learning methods

This section provides an overview on four different groups of enhanced machine learning methods already mentioned in Introduction. First, we introduce variable selection methods (forward, backward and stepwise selection). Second, we describe penalization methods (Lasso, Ridge and elastic net regression methods). Third, we explain the random forests method. Fourth, we introduce neural networks.

Variable selection

Variable selection[4] comprises the process of automatically selecting a subset of relevant variables in a data set. The main objective of variable selection methods is eliminating irrelevant or nearly redundant variables without losing too much information. Thereby, variable selection improves linear regression models by specifying a criterion to decide which variables should be included in or excluded from the model and helps to simplify forecasting models, thereby reducing their susceptibility to over-fitting problems. Furthermore, variable selection has a positive effect on training times, which is especially relevant in large data sets, and improves the interpretability of forecasting models. Variable selection is often used in data sets with multiple variables and comparatively few observations. The commonly used variations of this method in the literature are forward, backward and stepwise variable selection, which will, respectively, be introduced subsequently.

Forward variable selection starts with a model without any predictor and iteratively adds the most contributive predictors to the model. The algorithm stops when the improvement in the fit is no longer statistically significant. Forward selection is a greedy algorithm and does not reevaluate past solutions. An advantage of forward variable selection is that it can be applied even if the number of variables exceed the number of observations in a data set.

Backward selection starts with a model that contains all available variables and iteratively eliminates predictors with the smallest contribution to the fit of the model. Unlike forward selection, it can only be applied when the number of observations in the data set exceeds the number of variables. The algorithm terminates with a model when all the predictors are statistically significant.

Forward and backward selection can be combined to a hybrid stepwise selection algorithm. This algorithm starts with an initial model without any predictor variables and, like forward selection, sequentially adds the variable that yields the highest improvement in the model fit. After adding a new variable, the already included variable is reevaluated for removal from the model. Any variable that no longer provides an improvement in the model fit is deleted, as in backward selection.

For all three methods, the maximum number of variables $p_{\max}$ included in the model must be specified. Based on this specification, the algorithms, respectively, identify $p_{\max}$ different best models of different sizes. Subsequently, tenfold cross-validation is used to estimate the average root mean squared error (RMSE) for each of the $p_{\max}$ models. The RMSE is a commonly used measure of the forecast error. A mathematical definition follows in "Model comparison procedure" section. A tenfold cross-validation splits the data set into ten subsamples of approximately equal size. Subsequently, the model is fitted in-sample using nine of the subsamples as training data, and the RMSE is calculated for an out-of-sample forecast with the tenth subsample as the test data. This procedure is executed ten times and each subsample k ($k = 1, \ldots, 10$) is used once for the out-of-sample forecast. The repetition of the out-of-sample forecast with each of the ten subsamples as test sample and the remaining subsamples as training samples results in ten RMSE estimates, which are averaged. The average RMSE is used to compare the $p_{\max}$ models, and the model that minimizes the RMSE is selected as the forecasting model.

Penalized regression

Subsequently, we give a brief overview on penalized regression methods,[5] which are used to improve the forecasting accuracy of linear regression models that reveal a high variance of the model estimators. The problem of high variances of model estimators is the typically resulting high forecasting error of the model. Such problems are especially relevant in complex models that contain multiple variables and are susceptible to over-fitting. To reduce parameter variances, penalized regressions introduce constraints for limiting the model parameters. Lasso, Ridge and elastic net regressions are the three variations of penalized regression models that can all be used for this type of shrinkage of parameter variance. If $\beta = (\beta_1, \ldots, \beta_p)'$ defines the coefficient vector of the regression model, Lasso and Ridge regression restrict the

[4] Even though it often seems that enhanced machine learning methods have only been developed recently, variable selection was first introduced by Efroymson (1960). Also refer to Kuhn and Johnson (2016) and Hastie et al. (2017) for a comprehensive description of variable selection methods.

[5] Refer to Kuhn and Johnson (2016), Hastie et al. (2017) and Gu et al. (2020) for more information on penalized regression methods.

norm $||\beta||$ of β by a pre-specified tuning parameter $t > 0$, where t controls the amount of shrinkage that is applied to the estimates. The only difference between these two penalized regression models is the specific choice of the applied norm ||.||. Lasso (least absolute shrinkage and selection operator) is based on the L^1-norm $||.||_1$, that is, the sum of absolute values of the regression coefficients β_j. Ridge regression applies the L^2-norm $||.||_2$, namely the root of sum of squares of the regression coefficients β_j.[6]

The elastic net approach combines the constraints of Lasso and Ridge regression by using a linear combination $q \cdot ||.||_2 + (1-q) \cdot ||.||_2^2$ of the L^1-norm and the squared L^2 -norm. Consequently, parameter estimations of the penalized regression models result from the following optimization problems[7]:

$$\hat{\beta}^{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \leq t, \tag{2}$$

$$\hat{\beta}^{\text{Ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{s.t.} \quad \sum_{j=1}^{p} \beta_j^2 \leq t, \tag{3}$$

$$\hat{\beta}^{\text{elastic}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{s.t.} \quad q \cdot \sum_{j=1}^{p} |\beta_j| + (1-q) \cdot \sum_{j=1}^{p} \beta_j^2 \leq t, \tag{4}$$

where y stands for the dependent variable and x_j for the explanatory variables of the penalized regression model. N describes the number of observations, p the total number of explanatory variables, and q is a weighting parameter. All three optimization problems can be rewritten in the Lagrangian form, where the respective Lagrangian parameter $\lambda = \lambda(t)$ depends on the tuning parameter t and, consequently, can also be interpreted as a tuning parameter controlling the degree of shrinkage[8]:

$$\hat{\beta}^{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{5}$$

$$\hat{\beta}^{\text{Ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{6}$$

$$\hat{\beta}^{\text{elastic}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}. \tag{7}$$

It is noteworthy that Ridge regression shrinks all unconstrained coefficients β_j by a uniform factor, while Lasso shrinks some coefficients and sets other coefficients to zero (Tibshirani 1996). Thus, in addition to shrinkage, Lasso is a tool to select the relevant variables and drop the unimportant ones.

Random forests

The following section contains a short description of the random forests method.[9] Random forests can be used both for classification problems, that is, the prediction of discrete or categorical variables and for regression problems, namely the prediction of continuous variables.[10] For our empirical analysis, we focus on the random forests regression method. A random forest is a combination of de-correlated decision trees. The structure of a binary decision tree is exhibited in Fig. 1.

A decision tree consists of a root node and several interior and leaf nodes. Starting at the root node and continuing in the interior nodes, the tree establishes a decision logic that splits the data set into several subsets. Each split represents a yes–no question based on a single variable or a combination

[6] Typically, the constraint $||\beta||_2 \leq \tau$ (with $\tau > 0$) of the Ridge regression is written as $||\beta||_2^2 \leq \tau^2 =: t$ (with $t > 0$). Consequently, the constraint of the Ridge regression is based on the squared L_2-norm $||.||_2^2$.

[7] We assume x_{ij} to be standardized and without loss of generality $\bar{y}_i = 0$ [see, (Tibshirani 1996)]. Thus, the intercept of the regression can be neglected.

[8] The optimal value λ is computed via cross-validation (Karabatsos 2014).

[9] Please refer to Kuhn and Johnson (2016), Hastie et al. (2017) and Gu et al. (2020) for a more comprehensive description of the random forests method.

[10] Note that in the literature on machine learning, the term "regression problem" is frequently used to describe the problem of forecasting a continuous dependent variable (Kuhn and Johnson 2016; Hastie et al. 2017). This term must not be confused with the term "linear regression model" that describes some of the models used in the present article.

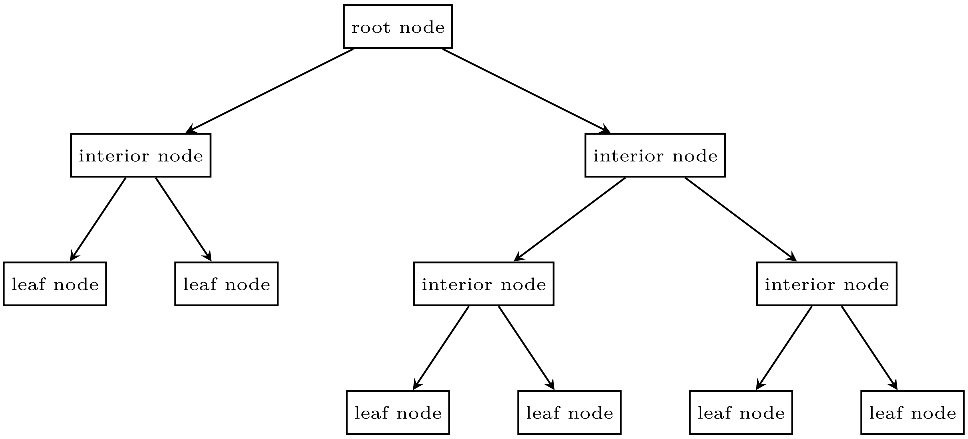


Fig. 1 A decision tree

of variables. The objective of the yes–no question in each node is to divide the data set into two subsets, which consist of observations that are more similar among themselves and different from the ones in the other subset, respectively. A central characteristic of a decision tree is the importance of the variables used in the tree. The impurity decrease that is reached through the division of the data into two subsets in a tree node is an indicator for the variable importance and, in regression trees, the impurity decrease can be measured by the variance of the dependent variable. The tree construction is stopped when the number of observations in a leaf node reaches a pre-specified minimum value. Generally, variables that are selected in the root node or in the upper interior nodes of the tree are more important than variables that are selected in the lower interior nodes of the tree because the upper splits lead to a greater decrease in impurity.

Decision trees can be advantageous over linear regression models because they allow non-parametric representations of the relationships between the dependent and explanatory variables. However, as linear regression models, decision trees pursue partially conflicting objectives. On the one hand, decision trees aim to minimize omitted variable biases, which can be achieved by trees that are grown very deep and are able to learn highly irregular patterns in a data set. On the other hand, deep grown trees tend to overfit the training data set, which can result in poor out-of-sample performance. The random forest algorithm introduced by Breiman (2001) intends to overcome this challenge. Therefore, it builds several de-correlated decision trees. Each tree is built over a random subsample drawn from a data set by iteratively selecting a random subset of the variables contained in that data set as described below. After the random forest algorithm has grown an ensemble of decision trees, an average is formed over all the decision trees to establish the random forest forecasting model. The out-of-sample forecast $\hat{f}_{\mathrm{rf}}^{B}$ of a random forest at a point x results as an average of all the individual forecasts from the single trees T_b $(b = 1, \dots, B)$:

$$\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x). \tag{8}$$

When constructing a random forest, one must determine the optimal number of trees B and the number of variables p_{rf} that are randomly selected and considered for each split in a tree. In the context of machine learning methods, such parameters are also referred to as hyperparameters. Typically, the optimal number of variables p_{rf} used in each tree is said to be $p/3$ for regression problems, where p is the overall number of explanatory variables (Probst et al. 2019).

A disadvantage of random forests is that the averaging over noisy but approximately unbiased trees introduce additional bias into the model and reduce the interpretability of results. However, a benefit of training the trees on different subsets of the training data set and selecting a random subsample of variables guarantees that the trees are de-correlated and, therefore, less prone to over-fitting. Through averaging across trees, the variance is also reduced, which generally leads to better forecasting performance. Therefore, random forests are one of the most popular machine learning algorithms. Additionally, random forests are invariant to scaling and various other transformations of variable values and robust to the inclusion of irrelevant variables. Furthermore, unlike linear regression, random forests do not require separate and prior variable selection, because their tree-based strategies automatically rank variables by their contribution to the decrease in impurity.[11]

Neural networks

Neural networks[12] comprise a large class of models that can be used for classification and regression problems. A neural

[11] In random forests, the decrease in impurity resulting from each variable can be averaged across the individual trees to determine the overall importance of that variable.

[12] Kuhn and Johnson (2016), Hastie et al. (2017) and Gu et al. (2020) provide a more detailed description on neural networks.

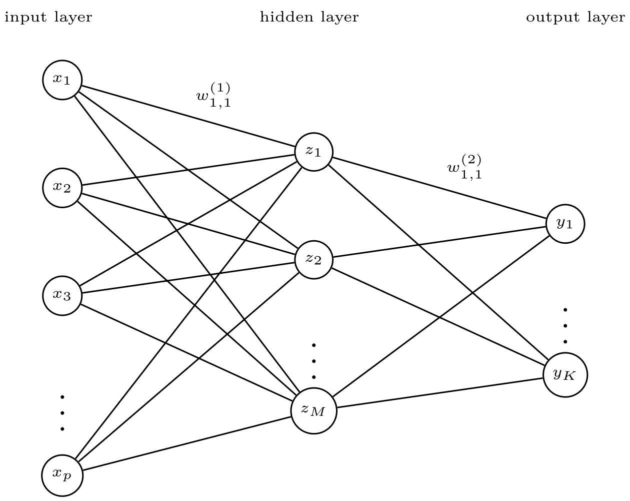


Fig. 2 A neural network

network consists of an input layer, one or more hidden layers and an output layer. Each layer has a set of nodes, which are called neurons. The nodes of the input layer represent the input variables of the model, whereas the nodes of the output layer represent the outputs. Empirical studies, such as the present study, usually aim to forecast the value of a dependent variable so that only one output is generated ($K = 1$). A set of edges connect the neurons of each layer are connected with the neurons of the subsequent layer. This structure is illustrated in Fig. 2, which shows a typical network with a single hidden layer.[13]

The values of the neurons in the hidden layer are calculated based on a function of the linear combinations of the input variable values, and the output value is generated as a linear combination of the neuron values in the hidden layer. More specifically, the values of the neurons in the hidden layer and the output layer are described by the following two equations:

$$\begin{aligned} z_m &= \sigma(w_{0,m} + w^{(1)'}_{i,m} x), \quad m = 1, \ldots, M, \quad i = 1, \ldots, p, \\ y_k &= w_{0,k} + w^{(2)'}_{m,k} z, \quad k = 1, \ldots, K, \quad m = 1, \ldots, M, \end{aligned} \tag{9}$$

with $x = (x_1, \ldots, x_p)$ and $z = (z_1, \ldots, z_M)$. The parameters $w^{(1)}_{i,m}$ and $w^{(2)}_{m,k}$ on the edges indicate the weight of neuron i from the input layer on neuron m in the hidden layer and neuron m from the hidden layer on neuron k in the output layer, respectively. Additionally, a bias node feeds into every node in the hidden layer and output layer, respectively. The bias nodes are captured by the intercepts $w_{0,m}$ and $w_{0,k}$. The so-called activation function σ is used to compute the values of the neurons in the hidden layer and to determine whether a neuron is activated or not. Commonly used activation functions are the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ and the hyperbolic tangent $\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x})$. Activated neurons in the hidden layer, which take a value in the interval [0, 1] (for the sigmoid activation function), respectively $[-1, 1]$ (for the hyperbolic tangent activation function), represent the inputs of the output layer. To obtain the final output—the dependent variable to be predicted—the activated neurons are multiplied by their respective weights and summed.

Following random initialization, the weights are iteratively adjusted through an algorithm minimizing the sum of squared errors. Various approaches can be used in this context; the most popular is the so-called backpropagation, which updates weights based on the gradient descent method. Therefore, the algorithm uses the partial derivatives of the sum of squared error function with respect to the weights. First, backpropagation determines a forecast of the output based on the initial weights. Subsequently, the error in the output layer is calculated and passed back through the hidden layer(s) to the input layer. In each layer, the errors are calculated with respect to the current weights. Based on the partial derivatives, the weights are updated through the gradient descent. This procedure guarantees approximation of the actual output through iterative model updates. An advantage of neural networks over linear models is their ability to describe (in coefficients) nonlinear relationships. In the empirical analysis, the sigmoid function is used for activation and a variation of the backpropagation algorithm[14] is used to fit the neural network.

Data

This section describes the data used in the empirical analysis. First, we explain the sample and the sample selection procedure. Second, we introduce the variables used in the analysis; third, we present descriptive statistics of the data.

[13] The exhibited network is a fully connected neural network, which means that a neuron of a certain layer is connected to all the neurons of the previous layer and vice versa. In applications, neural networks are usually fully connected (Hastie et al. 2017).

[14] We use the resilient backpropagation to fit our neural network models. This algorithm is similar to the common backpropagation algorithm described in "Neural networks" section but is usually faster and does not require a fixed learning rate (Naoum et al. 2012).

Sample selection

We use data on 597 CAT bonds issued between January 2002 and December 2017 for which we observe the premium at issue. The premia, which form the dependent variable in our analysis, can be described as yield spreads over the LIBOR. The data, which also include the bond ELs, issue volumes, and terms, are obtained from Aon Benfield. Data on the trigger mechanism, insured peril types, and locations are obtained from the Artemis Deal Directory and from Aon Benfield. Data on the bonds' sponsors are obtained from Lane Financial LLC and macroeconomic data are extracted from Bloomberg and Thomson Reuters.

We exclude all observations with missing or implausible data (e.g., observations where the EL does not equal the product of the PFL and the CEL). The final data used in the empirical analysis consist of 580 CAT bonds.

Variables

The set of variables included in the empirical analysis is based on the studies of Braun (2016) and Gürtler et al. (2016). We introduce both bond-specific variables and macroeconomic variables, which are described as follows.

Bond-specific variables

The set of bond-specific variables comprises the *EL*, which has the most important effect on premia. In addition, we establish the variable *Log(Volume)*, which represents the natural logarithm of a bond's issue volume and is a potential proxy for bond liquidity. The variable *Maturity* captures the impact of a bond's maturity on premia. Furthermore, to control for the effects of different trigger types on CAT bond premia, we establish a dummy variable *Trigger Indemnity*, which takes the value of one if the bond's trigger type is indemnity, and zero otherwise. The impact of the sponsor type on premia is modeled by introducing dummy variables for the sponsor types *Insurer*, *Reinsurer*, and *Other*. We control for a CAT bond's complexity by introducing the variables *No. of Locations* and *No. of Perils*, which measure the number of different insured locations/perils. Finally, we establish a series of dummy variables for different peril types, peril locations and bond rating categories, all of which are presented in Table 2.

Macroeconomic variables

Macroeconomic variables are used to consider the overall market development in our models. First, we measure the impact of the general development of prices on the CAT bond market. Therefore, we observe the volume-weighted mark-to-market price (weighted price) of outstanding CAT

Table 1 Summary statistics: cardinal CAT-bond-specific and macroeconomic variables

	Obs.	Mean	SD	Min.	Max.
CAT-bond-specific variables					
Premium (in %)	580	7.66	5.07	0.41	39.25
EL (in %)	580	2.57	2.56	0.00	17.35
Volume (in USD million)	580	132.45	122.01	1.80	1,500.00
Maturity (in month)	580	36.61	11.93	3.32	64.56
No. perils	580	1.85	1.12	1.00	5.00
No. locations	580	1.29	0.62	1.00	3.00
Macroeconomic variables					
Reins. Index	580	0.01	0.14	− 0.11	0.37
S&P500	580	0.01	0.03	− 0.11	0.12
Corp. spread	580	0.05	0.02	0.00	0.16
CAT bond Index	580	− 0.00	0.01	− 0.03	0.02

Statistics for all the variables are reported at the issue level

Table 2 Summary statistics: nominal and ordinal CAT bond-specific and macroeconomic variables

	Obs.	Percentage
Trigger		
Indemnity	223	38.45
Non-indemnity	357	61.55
Peril type		
EQ	378	65.17
HU	375	64.66
Wind	227	39.14
Other	35	6.03
Peril location		
EU	133	22.93
JP	89	15.34
NA	457	78.79
Latin America	46	7.93
Asia/Australia	23	3.97
Sponsor		
Insurer	258	44.48
Reinsurer	285	49.14
Other	37	6.38
Rating		
AA	4	0.69
A	4	0.69
BBB	19	3.28
BB	228	39.31
B	122	21.03
No rating	203	35.00

Statistics for all the variables are reported at the issue level. Since it is possible that multiple types of perils and locations are assigned to one bond, the category percentages do not add up to 100% for these groups of variables bonds on the secondary market. We then determine the relative change of that price on a monthly basis and label our variable *CAT Bond Index*. Note that CAT bonds are a potential substitute for traditional reinsurance, which suggests that the prices for those two types of risk transfer instruments show some co-movement (Braun 2016; Gürtler et al. 2016). For this reason, we incorporate the annual relative change in the Guy Carpenter Global Property Catastrophe Rate-on-Line Reinsurance Price Index (*Reins. Index*), which is described in more detail in Carpenter (2012). We use the change in that price index as a proxy for the reinsurance price cycle. Furthermore, we include the variable *Corporate Credit Spread (Corp. Spread)*, which is based on the credit spreads of US corporate bonds of different rating classes and maturities between one and 3 years, obtained from Bank of America Merrill Lynch. The variable *Corp. Spread* is constructed by matching the spreads with bonds in the identical rating class. Finally, we include the monthly return on the *S&P500* to model the development of equity markets.[15]

Descriptive statistics

This section presents descriptive statistics for the dependent and explanatory variables used for model construction. Table 1 presents the summary statistics on the cardinal variables applied in the empirical analysis. The variables are reported at the issue level. The mean of the premium is 7.66% and is almost three times greater than the mean of the EL. A bond has an average volume of 133 USD million and an average maturity of 37 months. On average a CAT bond insures 1.85 perils and 1.29 locations.

Table 2 shows the summary statistics of the nominal and ordinal variables used in the empirical analysis. Almost 40% of CAT bonds in our data set contain an indemnity trigger. Most bonds insure perils such as earthquakes (EQ) and hurricanes (HU). The most prevalent peril location is North America (NA). The sponsor type is “Reinsurer” for around 50% of the bonds, whereas the sponsor type is “Insurer” for 44% of the bonds. About 40% of the bonds have a “BB” rating.

Empirical analysis

In this section, we describe the procedure and the results of our empirical analysis. First, we outline our model comparison procedure and provide an overview of the considered models. Second, we present the hyperparameter tuning strategies applied to identify the best-performing models among the considered methods, respectively. Third we describe the out-of-sample results obtained with the different models and present a graphical analysis of the best-performing model subsequently.

Model comparison procedure

In the empirical analysis, we first establish a forecasting model for CAT bond premia using a linear regression based on the results obtained in the literature (Braun 2016; Gürtler et al. 2016). Apart from being more established in the asset pricing literature, a linear regression model has the advantage in that it enables us to assess whether the model structure and variables used are reasonable from an economic perspective, that is, more easily compared to random forests and neural network models. We use a three-step procedure to construct the linear benchmark model. First, we conduct an in-sample linear regression on the overall set of explanatory variables. Second, we eliminate statistically or economically insignificant variables from the model. The elimination is based on the variables’ p values ($p > 0.1$ for statistical significance) and standardized effects ($\beta < 0.1$ for economic significance). Third, we conduct the out-of-sample forecast using both the full and reduced linear models. To test the performance of our linear models and the subsequently introduced models, we conduct a rolling sample forecast, where we model the data in-sample over a 5-year period, and then use the model to conduct an out-of-sample forecast for the next year. After each forecast, we move forward the in-sample and out-of-sample periods by 1 year. As a result, we obtain a forecast for eleven periods over the time horizon in our data set.[16]

Next, we add the variable selection algorithms, namely forward, backward and stepwise variable selection and penalized regressions (Lasso, Ridge and elastic net) to the set of considered linear models. We compare the results obtained with the different linear models to the results obtained with a random forests model and a neural network model in terms of out-of-sample forecasting performance, based on the subsets described above. All out-of-sample forecasts obtained during the analysis are compared based on the RMSE as goodness-of-fit measure, which is defined as follows (Campbell and Thompson 2007; Xu and Taylor 1995):

$$\text{RMSE} = \sqrt{\frac{1}{N'}\sum_{i=1}^{N'}(prem_i - \widehat{prem_i})^2}, \tag{10}$$

[15] The macroeconomic variables are matched to the CAT bond data set based on the issue month of the respective bond.

[16] In additional analyses, we consider alternative lengths for the in-sample and out-of-sample periods.

Table 3 Hyperparameter choice for enhanced machine learning methods

Method	Hyperparameters	Grid search interval	Search increments	Cross-validated hyperparameter values
Linear regression with variable selection	Max. # variables p_{max}	[1, 26]	1	[8, 24]
Penalized regression	Shrinkage parameter λ	$[x_1 = 100, \dots, x_n = 0.0001, 0]$	$x_{i+1} = \frac{1}{10} \cdot x_i$	[0, 0.001]
RF	# split variables (p_{rf})	[1, 22]	1	[13, 21]
	# trees (B)	[100, 1000]	100	[100, 400]
NN	# hidden layers	[1, 3]	1	[2, 3]
	# neurons per hidden layer	[1, 26], [0, 26], [0, 26]	1	[1, 26], [1, 25], [0, 15]

This table presents the results of the hyperparameter tuning performed for the enhanced machine learning methods

where N' corresponds to the number of observations in the out-of-sample data, $prem_i$ is the observed risk premium of bond i, and $\widehat{prem}_i$ denotes the premium forecasted by the respective model.[17]

Hyperparameter tuning

Enhanced machine learning methods typically comprise one or multiple so-called hyperparameters, which prevent the machine learning algorithm from over-fitting the training data (Mullainathan and Spiess 2017). In order to find a model that provides good out-of-sample forecasts, it is necessary to first determine a suitable set of hyperparameter values, a process that is referred to as hyperparameter tuning (Mullainathan and Spiess 2017). There are two approaches to hyperparameter tuning. The first approach is to split the observed data set into three parts: training, validation and test sample. The considered model is then fit to the training sample with a specific set of hyperparameters. Subsequently, the validation set is used to tune the hyperparameters based on a suitable evaluation criterion. Finally, the optimal set of hyperparameters are used to test the actual out-of-sample performance in the test data set. The mentioned approach has the drawback of further reducing the size of data sets that are already small. A second alternative hyperparameter tuning approach is based on cross-validation in the training data set (Mullainathan and Spiess 2017). This approach provides a remedy for problems related with small sample size by using the training data more efficiently. Since we use a relatively small sample to conduct our analysis, we use tenfold cross-validation to tune the hyperparameters.

The determination of a well-performing set of hyperparameters requires a method to sample hyperparameter values and an objective function criterion for the model evaluation. Grid search and random search are the two most common approaches to sample hyperparameters. Grid search evaluates a predetermined set of hyperparameter values that result from the applied search grid. In comparison, random search chooses a random set of hyperparameter values to be evaluated at each cross-validation iteration. In our study, we sample hyperparameter values with grid search, in order to have control over the considered sets of hyperparameters. For each combination of hyperparameter values, we determine the average RMSE resulting from the cross-validation and subsequently select the hyperparameter set with minimum average RMSE.

Table 3 presents an overview of the hyperparameter tuning process for all the enhanced machine learning methods considered. Columns 1 and 2 show the method and the corresponding hyperparameters. Column 3 shows the grid search interval over which the hyperparameter values are tested. Column 4 presents the search increments applied to test the hyperparameter values within the search intervals. Column 5 exhibits the hyperparameter values chosen in the cross-validation procedure for the respective method.

For the variable selection methods (forward, backward and stepwise variable selection), we must specify the maximum number of variables $p_{\max}$ as described in "Variable selection" section. We test this parameter on the interval [1, 26] with an increment of 1. The lower bound of the interval would result in a model containing only one explanatory variable while its upper bound would result in a model containing all the explanatory variables in our data set. Depending on the variable selection method and the considered rolling sample period, the values selected for $p_{\max}$ range between 8 and 24. The three penalized regression methods (Ridge, Lasso and elastic net) require tuning for the shrinkage parameter λ, as described in "Penalized regression" section. We test the shrinkage parameters on the interval $[x_1 = 100, \dots, x_n = 0.0001, 0]$ and iteratively determine

[17] The RMSE is conventionally used as measure of the forecast error. This paper focuses on improving the forecasting performance for CAT bond premia on the primary market. To assess the performance of portfolio strategies on the secondary market, other measures, as for example the Sharpe ratio, could be taken into account.

Table 4 Out-of-sample forecasts

IS	OOS	Full	Reduced	Forward	Backward	Stepwise	Lasso	Ridge	Elastic net	RF	NN
2002–2006	2007	0.0171	0.0250	0.0179	0.0180	0.0178	0.0217	0.0181	0.0217	0.0157	0.0203
2003–2007	2008	0.0154	0.0246	0.0155	0.0152	0.0156	0.0154	0.0146	0.0154	0.0090	0.0154
2004–2008	2009	0.0184	0.0238	0.0194	0.0178	0.0194	0.0191	0.0188	0.0191	0.0075	0.0209
2005–2009	2010	0.0243	0.0286	0.0253	0.0245	0.0254	0.0263	0.0248	0.0263	0.0107	0.0246
2006–2010	2011	0.0156	0.0159	0.0160	0.0156	0.0156	0.0147	0.0153	0.0147	0.0068	0.0169
2007–2011	2012	0.0167	0.0242	0.0168	0.0167	0.0167	0.0168	0.0166	0.0177	0.0095	0.0205
2008–2012	2013	0.0200	0.0222	0.0219	0.0220	0.0227	0.0220	0.0206	0.0220	0.0101	0.0126
2009–2013	2014	0.0141	0.0153	0.0141	0.0203	0.0202	0.0155	0.0141	0.0155	0.0068	0.0119
2010–2014	2015	0.0118	0.0116	0.0126	0.0122	0.0127	0.0118	0.0115	0.0118	0.0051	0.0075
2011–2015	2016	0.0148	0.0166	0.0149	0.0149	0.0138	0.0145	0.0141	0.0145	0.0071	0.0092
2012–2016	2017	0.0111	0.0104	0.0113	0.0110	0.0167	0.0101	0.0109	0.0101	0.0071	0.0103
Mean		0.0163	0.0198	0.0169	0.0171	0.0179	0.0171	0.0163	0.0171	0.0087	0.0155
Median		0.0156	0.0222	0.0160	0.0167	0.0167	0.0155	0.0153	0.0155	0.0075	0.0154
SD		0.0037	0.0061	0.0041	0.0040	0.0038	0.0048	0.0041	0.0048	0.0029	0.0056

This table reports the RMSE of the out-of-sample estimation for the full linear regression model, which is based on Braun (2016) and Gürtler et al. (2016), the reduced model after the three-step selection, linear regression with variable selection, penalized regression, random forests (RF) and neural network (NN) models. The results, in terms of the RMSE, are presented for each rolling sample period. Additionally, the mean, median and standard deviation of the RMSE across all rolling sample periods are shown

the increments according to Table 3. For all three methods and the considered rolling sample periods, the parameter $\lambda = [0, 0.001]$ is optimal, in which $\lambda = 0$ only is optimal for a few rolling sample periods.[18]

In the random forests model, the number of variables p_{rf}, which are randomly sampled as candidates for each split and the number of trees B must be tuned. Probst et al. (2019) state that selecting an optimal parameter for the number of split variables is a trade-off between stability and accuracy of each tree. This is attributed to the fact that lower values of p_{rf} produce more de-correlated trees, which, aggregated to a random forest, lead to better stability. On the other hand, a higher value of p_{rf} may lead to trees that perform better on average, since trees with a low p_{rf} value are built on a possibly suboptimal set of variables. Dealing with this trade-off, we start with the default value of $p_{\text{rf}} = p/3$ for the regression and iteratively raise p_{rf} by 1 until the cross-validated value of p_{rf} for each rolling sample interval is smaller than the maximum grid search interval value. This procedure results in a test interval of split variables between 1 and 22 with an increment of 1, the optimal specification lies between 13 and 21 depending on the considered rolling sample period. For the number of trees, we test the interval [100, 1000] with an increment of 100, and the optimal specification—again independence of the considered rolling sample period—lies between 100 and 400.

For the neural network, the number of hidden layers and neurons in each hidden layer must be tuned. For the number of neurons, the interval [1, 3] is tested, because the literature suggests that performance of the neural network declines with more than 3 hidden layers in small data sets (Gu et al. 2020; Arnott et al. 2019). Our tuning suggests that 3 hidden layers provide the optimal specification for all the considered rolling sample periods, except the fifth rolling sample period where 2 hidden layers turn become the optimal model specification. For the number of neurons per hidden layer, the intervals [1, 26] (first hidden layer), [0, 26] (second hidden layer) and [0, 26] (third hidden layer) are tested. Depending on the considered rolling sample period, the optimal network designs differ widely over the intervals [1, 26] (first hidden layer), [1, 25] (second hidden layer) and [0, 15] (third hidden layer).

Out-of-sample results

In this section, we present the results of our empirical analysis. Table 4 shows the performance of the considered linear regression and enhanced machine learning models in the out-of-sample forecast.[19] Columns 1 and 2 report the considered in-sample and out-of-sample periods of the rolling

[18] The range chosen for the grid search interval in our analysis is inspired by Zou and Hastie (2003).

[19] In-sample results are available with the authors and can be provided on request.

Table 5 Out-of-sample forecasts with 2-year out-of-sample period

IS	OOS	Full	Reduced	Forward	Backward	Stepwise	Lasso	Ridge	Elastic net	RF	NN
2002–2006	2007–2008	0.0196	0.0268	0.0208	0.0207	0.0209	0.0218	0.0198	0.0218	0.0245	0.0317
2003–2007	2008–2009	0.0204	0.0254	0.0201	0.0202	0.0202	0.0204	0.0193	0.0204	0.0137	0.0182
2004–2008	2009–2010	0.0350	0.0345	0.0323	0.0307	0.0323	0.0307	0.0338	0.0307	0.0275	0.0387
2005–2009	2010–2011	0.0220	0.0236	0.0223	0.0224	0.0225	0.0227	0.0221	0.0227	0.0154	0.0226
2006–2010	2011–2012	0.0232	0.0197	0.0228	0.0233	0.0232	0.0184	0.0203	0.0184	0.0160	0.0255
2007-2011	2012–2013	0.0213	0.0281	0.0209	0.0213	0.0213	0.0210	0.0209	0.0214	0.0209	0.0305
2008–2012	2013–2014	0.0215	0.0224	0.0225	0.0222	0.0231	0.0227	0.0220	0.0227	0.0194	0.0312
2009–2013	2014–2015	0.0147	0.0156	0.0152	0.0194	0.0192	0.0158	0.0146	0.0158	0.0144	0.0217
2010–2014	2015–2016	0.0174	0.0169	0.0178	0.0177	0.0180	0.0174	0.0170	0.0174	0.0162	0.0306
2011–2015	2016–2017	0.0257	0.0244	0.0258	0.0258	0.0211	0.0229	0.0240	0.0229	0.0263	0.0312
Mean		0.0221	0.0237	0.0221	0.0224	0.0222	0.0214	0.0214	0.0214	0.0194	0.0282
Median		0.0214	0.0240	0.0216	0.0218	0.0212	0.0214	0.0206	0.0216	0.0178	0.0306
SD		0.0055	0.0056	0.0046	0.0037	0.0039	0.0041	0.0051	0.0041	0.0051	0.0061

This table reports the RMSE of the out-of-sample estimation for the full linear regression model, which is based on Braun (2016) and Gürtler et al. (2016), the reduced model after the three-step selection, linear regression with variable selection, penalized regression, random forests (RF) and neural network (NN) models. The results, in terms of the RMSE, are exhibited for each rolling sample period. Additionally, the mean, median and standard deviation of the RMSE across all rolling sample periods are shown. A 2-year period is used for the out-of-sample forecast

sample analysis. The following columns show the performance of all the models considered in terms of RMSE.[20]

The full linear regression model based on Braun (2016) and Gürtler et al. (2016) has a mean RMSE of 0.0163. In comparison, the reduced model has a higher RMSE of 0.0198 and a considerably higher standard deviation. All three variable selection models presented in the following columns exhibit similar means, medians and standard deviations of the RMSE. Forward selection exhibits the lowest mean (median) RMSE but has the highest standard deviation compared to backward and stepwise selection. Notably, the Ridge regression performs best among the penalization methods while the Lasso and elastic net methods lag behind, but perform equally well in the mean, median and standard deviation of the RMSE. The Ridge model produces the same mean RMSE as the full linear regression model and even a lower median *RMSE*, nonetheless the Ridge model has a higher standard deviation of the RMSE. As the mean (and median) performance is (almost) equal, one would prefer the more stable full linear regression model. Overall, neither the variable selection methods nor the penalization methods perform better than the full linear regression models in terms of the mean, median and standard deviation of the RMSE.

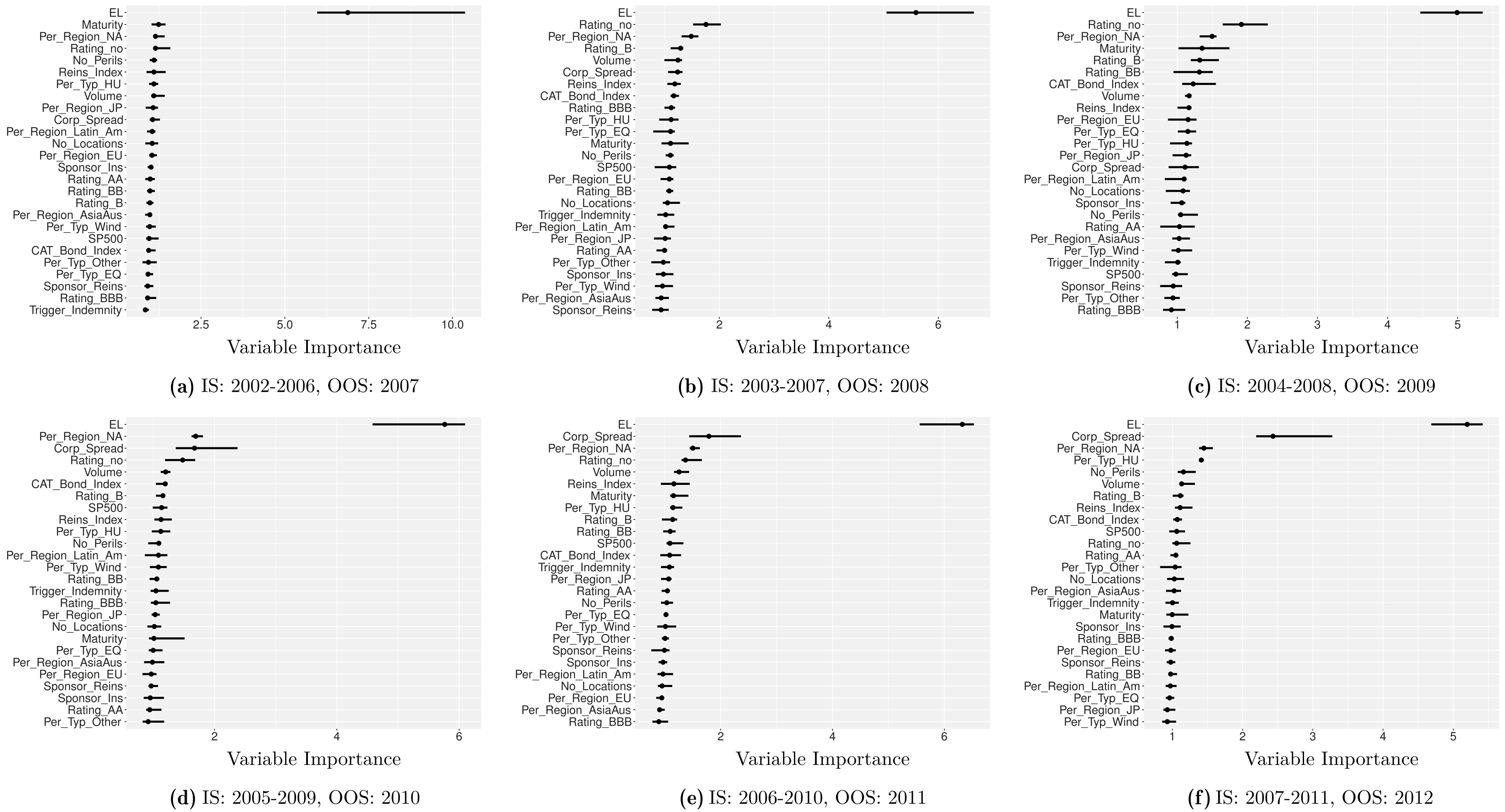
The random forest yields a considerably lower mean (median) RMSE of 0.0087 (0.0075) than all the considered linear regression models and, additionally, presents a lower standard deviation. With a mean (median) RMSE of 0.0155 (0.0154), the neural network slightly outperforms, all the linear regression models except the Ridge regression, but it continues to lag the random forest. In addition, apart from the reduced linear regression model, the neural network exhibits the highest standard deviation of the RMSE, which raises concerns about the stability of the neural network forecasts over time.

Broadly, the performance of the random forest indicates that enhanced machine learning methods can outperform linear regression models in relatively small data sets. However, the relatively weak performance of the neural network shows that even with a systematic choice of hyperparameters, overfitting remains a challenging issue, when applying enhanced machine learning methods, especially in small data sets.

Next, we extend the out-of-sample period from 1 to 2 years and conduct another test of our models' performance. The results shown in Table 5 support our previous results.[21] The random forest exhibits the lowest mean (median) RMSE. The neural network performs worse than all other models. While the mean (median) forecasting performance of the random forest is significantly better than the mean performance of the linear regression, the standard deviations are nearly the same. Therefore, based on the changed test setting, the random forest can be considered as the best performing forecasting model. However, the performance difference between random forest and the linear regression models decreases, which may indicate that the random forest needs more data to fit the model when the forecasting horizon increases.

[20] Results based on R^2_{OS} are qualitatively identical and are available with the authors on request. Refer to Campbell and Thompson (2007) and Xu and Taylor (1995) for a definition of the R^2_{OS}.

[21] The results based on R^2_{OS} are qualitatively identical and are available on request.



(a) IS: 2002-2006, OOS: 2007

(b) IS: 2003-2007, OOS: 2008

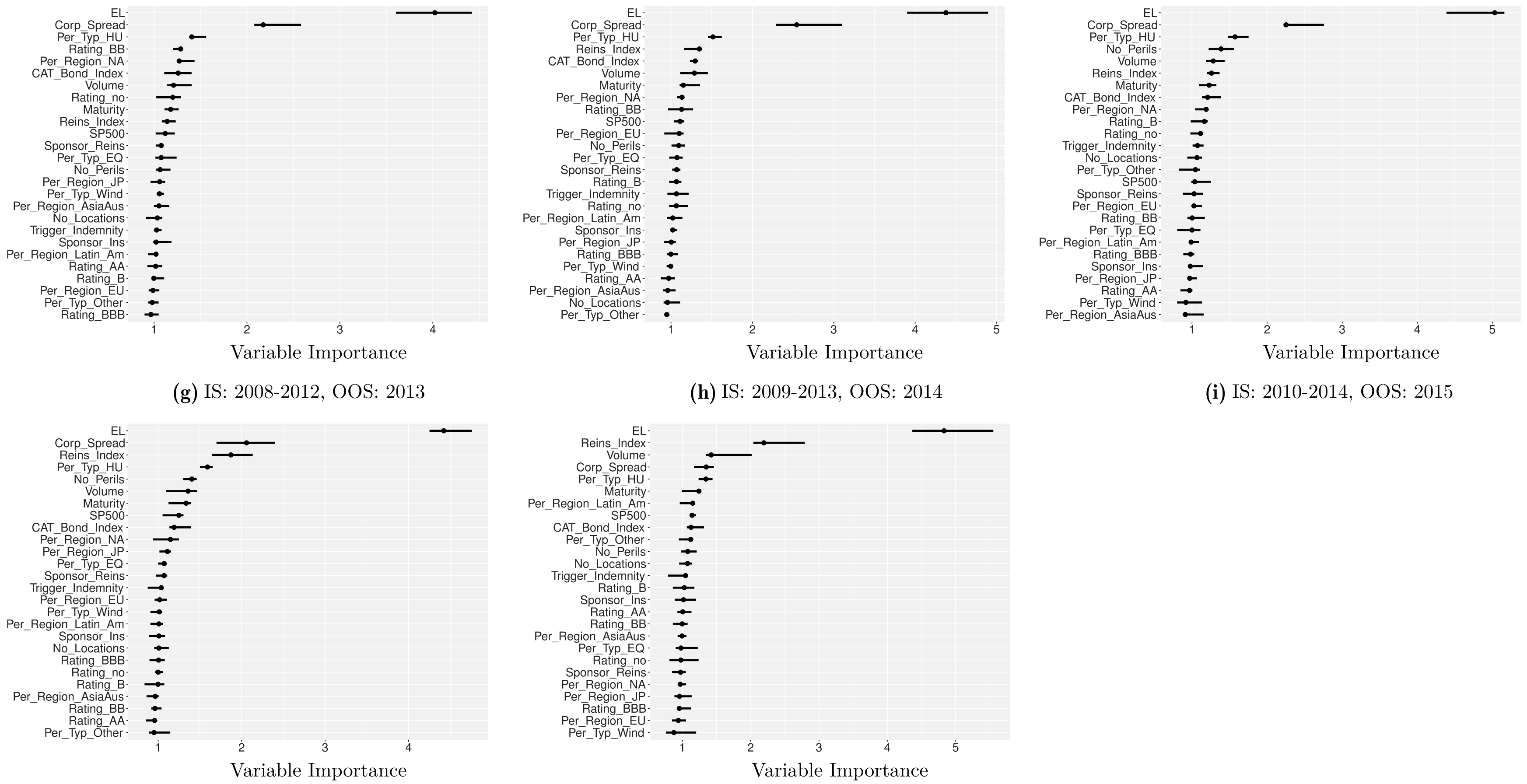
(c) IS: 2004-2008, OOS: 2009

(d) IS: 2005-2009, OOS: 2010

(e) IS: 2006-2010, OOS: 2011

(f) IS: 2007-2011, OOS: 2012

Fig. 3 Variable importance given by RF. This figure shows the importance of the variables used in the random forest model. For each variable, the horizontal line presents the range of the factor by which the model’s forecasting error increases due to a permutation of the variable. The black dot on the horizontal line represents the median importance of a variable



(g) IS: 2008-2012, OOS: 2013

(h) IS: 2009-2013, OOS: 2014

(i) IS: 2010-2014, OOS: 2015

(j) IS: 2011-2015, OOS: 2016

(i) IS: 2012-2016, OOS: 2017

Fig. 3 (continued)

Graphical model analysis

Based on the results of the prior literature, we know that the EL is the most influential determinant of CAT bond premia (Galeotti et al. 2013; Braun 2016; Gürtler et al. 2016; Trottier et al. 2018). The random forest model confirms the dominant influence of the EL, which can be observed in Fig. 3, where we depict the importance of the variables used in the model. In this figure, the horizontal line presents the range of the factor by which the model's forecasting error increases when the variable is permuted. Thus, the larger the increase in forecasting error measured by the RMSE over all the trees of the random forest, the more important will be the variable. The black dot on the respective horizontal line represents the median importance of a variable aggregated over the trees.

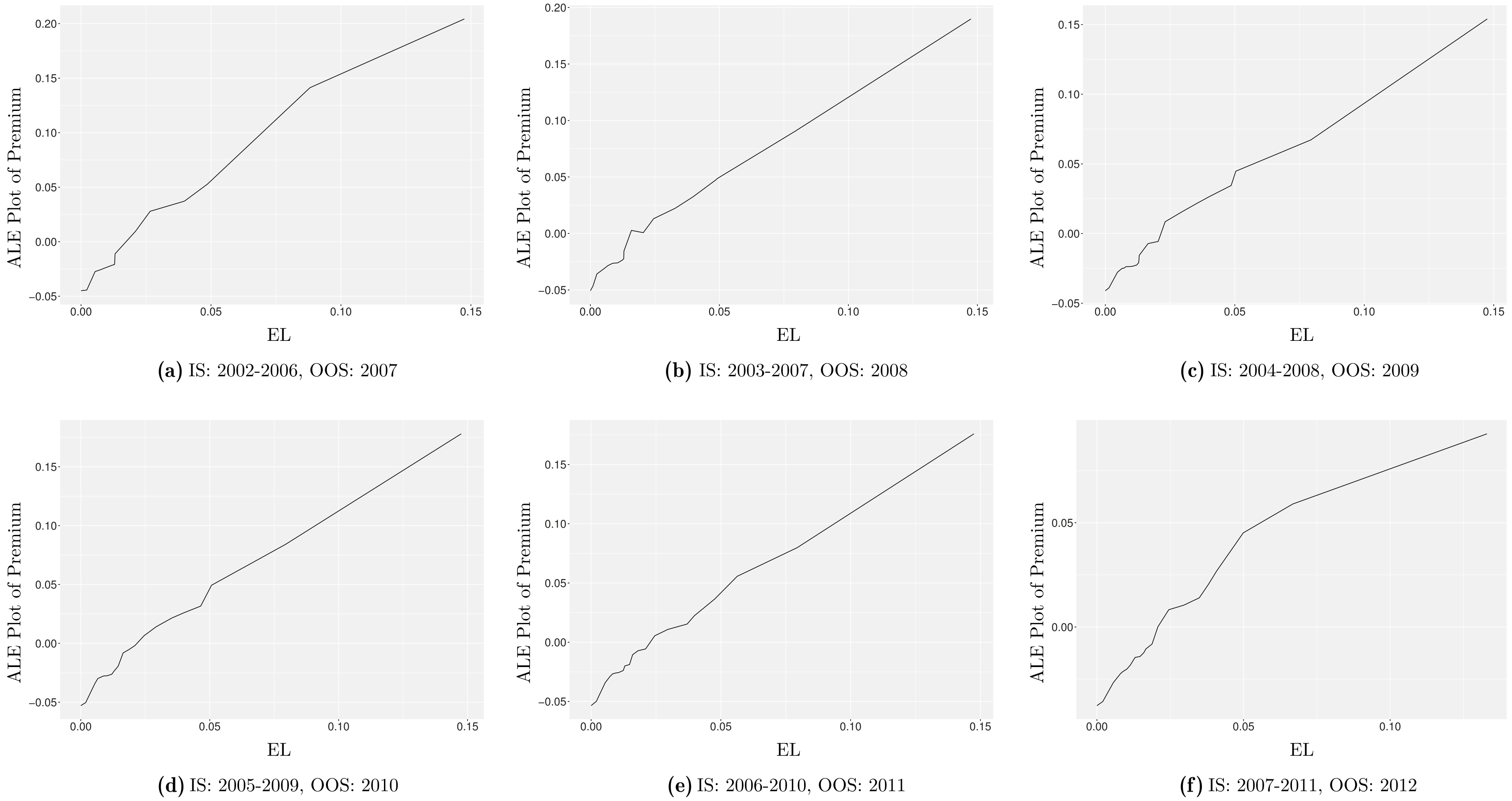
Consequently, it is important to understand which relationship between the EL and the CAT bond premium is adopted by our best-performing model, the random forest. In this section, we present this relationship graphically. Such a graphical presentation of the model structure is particularly helpful for the random forests model, because its structure, that is, the relationship between the dependent and explanatory variables, cannot be understood based on a set of coefficients as is the case for linear regression models. To develop a graphical representation, we construct accumulated local effect (ALE) plots for each rolling sample period. The ALE plots describe how a variable (in this case, we consider the EL) influences the forecast of a machine learning model, on average. The ALE plot exhibits the local effect of a variable within a certain interval. To calculate the ALE, the EL is divided into 20 intervals based on the quantiles of the EL-distribution. Subsequently, the local effects of the EL are computed by determining the differences in premium forecasts between each consecutive pair of observations within the interval, which are sorted in ascending order by their respective ELs. Next, all the local effects within an interval are averaged. Finally, the ALEs are accumulated to the connected graphs as can be seen in Fig. 4. The ALE plots are centered so that the mean effect is zero.

In Fig. 4, the graphs present an overall concave structure, and, not surprisingly, a positive relationship between the premium and EL. However, while the slope of the graphs is seen decreasing toward the higher values of the EL, locally, it exhibits convex areas, where the premium increases more sharply with the increasing EL. Thus, the non-trivial relationship between the premium and EL in Fig. 4 suggests that it is challenging to identify a parametric representation of the premium as a function of the EL as would be required by a linear regression or a neural network model. Therefore, Fig. 4 provides a potential explanation for the random forest outperforming the other considered methods.

Conclusion

This study assesses the potential of enhanced machine learning methods to improve forecasting models in asset pricing. The tests conducted for the forecast of CAT bond premia show that extensions of the linear regression model based on variable selection or penalization methods exhibit relatively small performance differences in comparison with the full linear regression model and will not improve the forecast produced by the full model. The mean (median) forecasting performance of the random forest is substantially higher than that of the linear regression models and the neural network; furthermore, the forecasting performance of the random forest exhibits a smaller standard deviation. This result is also supported in an alternative analysis with a 2-year out-of-sample period. Overall, enhanced machine learning methods seem to have the potential to improve the forecasts of CAT bond premia and prices of other asset classes, even if only a relatively small data set is available. In the context of our application, random forests provide stable and significant performance improvements, whereas the performance of neural networks is very unstable compared to the performance of other models. Especially, against the background of uncertain future market conditions, the stability of the forecasts of random forests is a valuable result.

This study has two main contributions to the literature on asset pricing. First, our analysis of the performance of different machine learning methods in forecasting CAT bond premia provides an indication of their potential for asset pricing, which is relevant for scientific literature and practitioners who consider the use of machine learning methods. In this context, a valuable finding of our study is that random forests can perform effectively and stable for a relatively small asset class. Second, a central explanation for the superior performance of random forests is attributable to the fact that they model the CAT bond premium without a specific distribution assumption. Thus, our study provides evidence that enhanced machine learning methods enable forecasting model improvements even in markets where the influencing variables are essentially known.



(a) IS: 2002-2006, OOS: 2007

(b) IS: 2003-2007, OOS: 2008

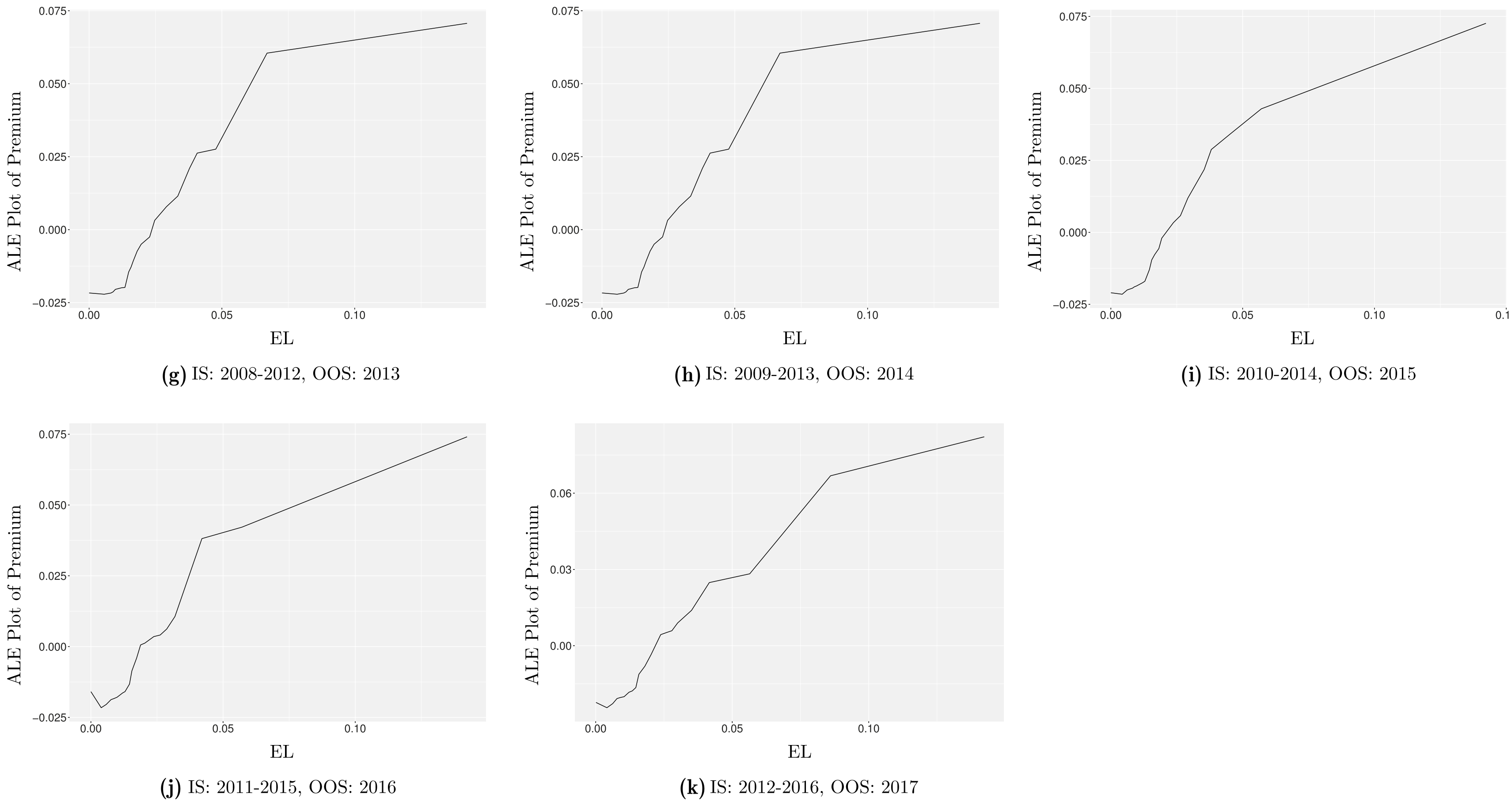
(c) IS: 2004-2008, OOS: 2009

(d) IS: 2005-2009, OOS: 2010

(e) IS: 2006-2010, OOS: 2011

(f) IS: 2007-2011, OOS: 2012

Fig. 4 CAT bond premium as a function of the EL. This figure shows the ALE plot of the CAT bond premium as a function of the EL. The plot describes the impact of the EL on the random forest model forecast for each rolling sample period



(g) IS: 2008-2012, OOS: 2013

(h) IS: 2009-2013, OOS: 2014

(i) IS: 2010-2014, OOS: 2015

(j) IS: 2011-2015, OOS: 2016

(k) IS: 2012-2016, OOS: 2017

Fig. 4 (continued)

Acknowledgements Open Access funding provided by Projekt DEAL. We thank the Gesamtverband der deutschen Versicherungswirtschaft e.V. for funding this research project.

References

Arnott, R., C.R. Harvey, and H. Markowitz. 2019. A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science* 1(1): 64–74.

Bongaerts, D., K.M. Cremers, and W.N. Goetzmann. 2012. Tiebreaker: Certification and multiple credit ratings. *The Journal of Finance* 67(1): 113–152.

Boot, A.W., T.T. Milbourn, and A. Schmeits. 2005. Credit ratings as coordination mechanisms. *The Review of Financial Studies* 19(1): 81–118.

Braun, A. 2016. Pricing in the primary market for cat bonds: New empirical evidence. *Journal of Risk and Insurance* 83(4): 811–847.

Breiman, L. 2001. Random forests. *Machine Learning* 45(1): 5–32.

Campbell, J.Y., and S.B. Thompson. 2007. Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies* 21(4): 1509–1531.

Cantor, R., and F. Packer. 1997. Differences of opinion and selection bias in the credit rating industry. *Journal of Banking and Finance* 21(10): 1395–1417.

Carpenter, G. 2012. *Catastrophes, cold spots and capital: Navigation for success in a transitioning market*. New York: Guy Carpenter & Company.

Cummins, J.D., and M.A. Weiss. 2009. Convergence of insurance and financial markets: Hybrid and securitized risk-transfer solutions. *Journal of Risk and Insurance* 76(3): 493–545.

Efroymson, M. 1960. Multiple regression analysis. In *Mathematical methods for digital computers*, ed. A. Ralston, and H.S. Wilf, 191–203. New York: Wiley.

Froot, K.A. 2001. The market for catastrophe risk: A clinical examination. *Journal of Financial Economics* 60(2–3): 529–571.

Galeotti, M., M. Gürtler, and C. Winkelvos. 2013. Accuracy of premium calculation models for cat bonds: an empirical analysis. *The Journal of Risk and Insurance* 80(2): 401–421.

Gron, A. 1994. Capacity constraints and cycles in property-casualty insurance markets. *The RAND Journal of Economics* 25(1): 110–127.

Gu, S., B. Kelly, and D. Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33(5): 2223–2273.

Gürtler, M., M. Hibbeln, and C. Winkelvos. 2016. The impact of the financial crisis and natural catastrophes on cat bonds. *Journal of Risk and Insurance* 83(3): 579–612.

Hastie, T., R. Tibshirani, and J.H. Friedman. 2017. The elements of statistical learning: Data mining, inference, and prediction. Springer series in statistics. Springer, New York, NY, second edition, corrected at 12th printing 2017 edition.

Karabatsos, G. 2014. Fast marginal likelihood estimation of the ridge parameter(s) in ridge regression and generalized ridge regression for big data. arXiv preprint arXiv:1409.2437.

Khandani, A.E., A.J. Kim, and A.W. Lo. 2010. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance* 34(11): 2767–2787.

Kuhn, M., and K. Johnson. 2016. *Applied predictive modeling*, corrected 5 printing ed. New York: Springer.

Lane, M. 2018. Pricing cat bonds: Regressions and machine learning: Some observations, some lessons. *Working paper, Lane Financial LLC*.

Lane, M. 2000. Pricing risk transfer transactions. *Astin Bulletin* 30(2): 259–293.

Makariou, D., P. Barrieu, and Y. Chen. 2020. A random forest based approach for predicting spreads in the primary catastrophe bond market. arXiv preprint arXiv:2001.10393.

Mullainathan, S., and J. Spiess. 2017. Machine learning: An applied econometric approach. *Journal of Economic Perspectives* 31(2): 87–106.

Naoum, R.S., N.A. Abid, and Z.N. Al-Sultani. 2012. An enhanced resilient backpropagation artificial neural network for intrusion detection system. *International Journal of Computer Science and Network Security (IJCSNS)* 12(3): 11.

Probst, P., M.N. Wright, and A.-L. Boulesteix. 2019. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9(3): e1301.

Rapach, D.E., J.K. Strauss, and G. Zhou. 2010. Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies* 23(2): 821–862.

Skreta, V., and L. Veldkamp. 2009. Ratings shopping and asset complexity: A theory of ratings inflation. *Journal of Monetary Economics* 56(5): 678–695.

Thornton, D.L., and G. Valente. 2012. Out-of-sample predictions of bond excess returns and forward rates: An asset allocation perspective. *The Review of Financial Studies* 25(10): 3141–3168.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society* 58(1): 267–288.

Trottier, D.-A., V.-S. Lai, and A.-S. Charest. 2018. Cat bond spreads via hara utility and nonparametric tests. *The Journal of Fixed Income* 28(1): 75–99.

Wang, S.S. 2004. Cat bond pricing using probability transforms. *Geneva Papers: Etudes et Dossiers, Special Issue on "Insurance and the State of the Art in Cat Bond Pricing"* 278: 19–29.

Wang, S.S. 2000. A class of distortion operators for pricing financial and insurance risks. *Journal of Risk and Insurance* 67(1): 15–36.

Xu, X., and S.J. Taylor. 1995. Conditional volatility and the informational efficiency of the phlx currency options market. *Journal of Banking and Finance* 19(5): 803–821.

Zou, H., and T. Hastie. 2003. Regression shrinkage and selection via the elastic net, with applications to microarrays. *Journal of the Royal Statistical Society Series B (Statistical Methodology)* 67: 301–20.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Tobias Götze graduated with a Master in industrial engineering from Braunschweig Institute of Technology, Germany, in 2014. Afterwards, he joined the Department of Finance at Braunschweig Institute of Technology, as a Ph.D. candidate and research assistant. Since 2020, he has been working with the reinsurer Hannover Re. His research activities are focused on the transfer of catastrophic risks to the capital market and the pricing of catastrophe bonds.

Marc Gürtler graduated in mathematics from the University of Cologne, Germany, in 1992. Subsequently, he worked as a risk manager at AXA Colonia Insurance Company. In 1995, he joined the Department of Business Administration at the University of Bonn, Germany, as a Ph.D. candidate and research assistant. He received the Ph.D. degree from the University of Bonn in 1997. From 1997 to 2002, he was an Assistant Professor in the Departments of Business Administration at the University of Bonn and at RWTH Aachen University. He received the habilitation degree in Business Administration from RWTH Aachen University in 2002. Since 2002, he has been a full Professor of Finance at Braunschweig Institute of Technology. His research interests are focused on the pricing of risk transfer instruments and different types of labor market signaling.

Eileen Witowski graduated in financial and business mathematics from Braunschweig Institute of Technology, Germany, in 2018. Afterwards, she joined the Department of Finance at Braunschweig Institute of Technology as a Ph.D. candidate and research assistant. Her research activities are focused on machine learning in insurance.