RESEARCH

Swiss Finance Institute
Research Paper Series
N°21-95

Machine Learning for Predicting Stock Return Volatility

Damir Filipović
Ecole Polytechnique Fédérale de Lausanne and Swiss Finance Institute

Amir Khalilzadeh
Ecole Polytechnique Fédérale de Lausanne

Machine Learning for Predicting Stock Return Volatility

Damir Filipovic[*] and Amir Khalilzadeh[†]

December 23, 2021

Abstract

We use machine learning methods to predict stock return volatility. Our out-of-sample prediction of realised volatility for a large cross-section of US stocks over the sample period from 1992 to 2016 is on average 44.1% against the actual realised volatility of 43.8% with an R^2 being as high as double the ones reported in the literature. We further show that machine learning methods can capture the stylized facts about volatility without relying on any assumption about the distribution of stock returns. Finally, we show that our long short-term memory model outperforms other models by properly carrying information from the past predictor values.

Keywords: Volatility Prediction, Volatility Clustering, LSTM, Neural Networks, Regression Trees.

JEL Classification: C51, C52, C53, C58, G17.

[*] Ecole Polytechnique Fédérale de Lausanne and Swiss Finance Institute, CH 1015 Lausanne, Switzerland. Damir.Filipovic@epfl.ch.

[†] Ecole Polytechnique Fédérale de Lausanne and EPFL Extension School, CH 1015 Lausanne, Switzerland. Amir.Khalilzadeh@epfl.ch.

1 Introduction

Volatility or fluctuation refers to the changes in a sequence of observations. In financial economics, it can be thought of as the standard deviation of the random term in a continuous-time diffusion model. As noted by Campbell et al. (1997), problems in financial economics would be mere exercises in basic microeconomics in the absence of uncertainty. In particular, asset return volatility is central to the theory and practice of asset pricing, asset allocation and risk management. In financial risk management, one uses forecasts of volatility to calculate the portfolio risk measures such as value-at-risk and expected shortfall. In the context of option pricing, the assumption of constant volatility generates a deviation between the theoretical prices and the actual market prices and thus one needs an accurate forecast of the underlying's volatility over the life of the option. Finally, investors can engage in volatility arbitrage by calculating volatility risk premium based on the forecast of volatility.

Despite its simple definition, volatility has been a challenging notion in financial economics as it is not directly observable. Our lack of knowledge about the level of uncertainty of asset returns poses great challenges on the daily decision making of practitioners as well as on academic research in financial economics. There are two main approaches one can estimate volatility using conditioning information.[1] In the first approach (hereinafter *time series volatility models, TSV*), one can estimate conditional volatility of excess stock returns through a parametric volatility model, such as (G)ARCH or stochastic volatility.[2] In the second approach (hereinafter *linear realized volatility models, LRV*), realized volatility for a

[1] There are at least two other approaches for this purpose that are less prevalent. One can regress excess returns onto a set of conditioning variables. The resulting squared residuals then will be regressed onto the same set of conditioning variables. The conditional variance will be the fitted values from the second regression. See for example Campbell (1987) and Breen et al. (1989). This approach has not been used extensively and does not provide superior advantages over the other methods. Also, one can use the option implied standard deviation as a forecast for volatility. Such forecasts are biased because most of the option pricing models do not allow a premium for bearing volatility risk. Moreover, as noted by Poon and Granger (2003) a test for such forecasts is a joint test of options market efficiency and a correct option pricing model. Lastly, option prices are not available for all assets, and such a framework does not allow one to find the causes of volatility changes. Finally, we do not discuss here methods that predict volatility based on past standard deviations, e.g. exponentially weighted moving average (EWMA) model.

[2] See the seminal works of Engle (1982) and Bollerslev (1986) where the class of Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models were pioneered. See Andersen et al. (2006) for a survey of GARCH and stochastic volatility models.

certain frequency (e.g. monthly) can be calculated using prices at a higher frequency (e.g., daily).[3] The projection of this measure of volatility onto an conditioning information set results in conditional volatility. That is, one can treat volatility as observed, and use a simple linear forecasting model to predict the realized volatility of asset returns. Engle and Patton (2001) outline the stylized facts about volatility that a good volatility model should be able to capture. These are volatility persistence and mean-reversion, leverage effect (asymmetric impact of innovations), heavy-tailed distribution of returns and the possibility of other predictors influencing volatility. These requirements pose challenges on each of the two methods described above.

TSV models proved to be very successful in capturing most of the stylized facts about volatility. However, they are not able to account for a rich conditioning information set. Numerous predictors (e.g. firm characteristics and macroeconomic variables) and their multiway interactions, which can amount to thousands, might have predictive power for the volatility. LRV models, on the other hand, have the advantage of using realized volatility as well as accommodating a larger set of predictors in a linear setting. However, as discussed by Paye (2008) such framework is subject to issues such as biasedness and inconsistency of estimated coefficients. Furthermore, both types of models are heavily confined by their parametric specifications. Finally and most importantly, TSV and LRV models are shown to have poor out-of-sample performance.

In this paper, we propose machine learning methods as a third approach for modelling and forecasting asset return volatility and argue that they are well suited in this context and in particular meet the criteria of being a good volatility model as outlined above. We show that these methods have the advantage of TSV models in capturing the stylized facts about volatility without being subject to distributional assumptions as the TSV models do. Also,

[3] Andersen et al. (2003a) and Andersen et al. (2003b) using the theory of quadratic variation argue that realized volatility as a nonparametric measure is an unbiased estimator of actual volatility as it is free of parametric assumptions of volatility models. It provides a consistent estimate of ex-post return variability and is superior to parametric GARCH or stochastic volatility models at capturing volatility. Also, an important advantage of using realized volatility is that it serves as an empirical ex-post benchmark for evaluating the quality of ex-ante volatility forecasts.

we show that machine learning methods can generalize the LRV models into a nonlinear setting and accommodate a large set of predictors and their multi-way interactions.

Machine learning in simple terms refers to a set of tools and techniques that requires minimal interference of humans in analyzing data. Instead, they allow computers to learn about the inherent structure in data through experience. The environment appears to them as a hierarchy of concepts, with each concept defined through its relation to simpler concepts. These advances, however, would not have been possible without the increasing availability of powerful and fast computing facilities. Machine learning methods have been widely used in various domains of science and technology and unlike the volatility models discussed above, these methods come with universal loss functions with no reliance on distributional assumptions and can accommodate a large number of predictors and their multi-way interactions.[4]

We examine the ability of machine learning methods in predicting realized volatility of stock return. In particular, we use gradient boosted regression trees and feedforward neural networks to predict realized volatility of stock return. We also develop an elastic net framework that essentially captures the linear relation between the predictors and the realized volatility. We let this model represent the LRV models previously used in the literature.[5] We proceed in two steps. First, we establish the superiority of our nonlinear models based on their out-of-sample performance, and further investigate whether they can also capture the stylized facts about the volatility of asset returns. To further explore the ability of the models in capturing volatility persistence, we deploy a long short-term memory

[4] David and Veronesi (2013) argue that investors observe and learn about the true economic environment and thus update their belief in a dynamic manner about movements of asset prices. They argue that we, as econometricians, do not have full information about the signals used by market participants to form their beliefs and thus make our predictions based on the information available to us. In this paper, we believe that machine learning methods, by learning the dynamics and relations between a large number of predictors in a very complex way, can replicate many strategies including those taken by investors when adjusting their belief about the movement of asset prices. They could help us understand the way investors analyse the true information.

[5] We do not necessarily aim to use the same variables and reproduce the results of the LRV models in the literature for two reasons. Our objective in this paper is not to provide a comparative analysis of various models. Instead, we aim to contrast the idea of using a linear framework and a general and flexible mapping between the predictors and the volatility such as those of machine learning methods. Secondly, almost all of the research done on this context is at the market level (market volatility versus economic variables) rather than individual stocks as we do (cross-section of stocks volatility versus firm characteristics and economic variables).

(LSTM) model and show that a flexible function of returns and volatility up to one year in the past produces an out-of-sample performance as good as an LSTM model with a full set of predictors. One can interpret such a model as a GARCH type model but with a flexible non-parametric functional form.

Our contributions to the literature are twofold. First, we demonstrate that machine learning methods have a great potential for predicting risk in the stock market. Most importantly, they can account for the stylized facts about the stock return volatility. They provide explainable results for the volatility clustering as well as the asymmetric impact of returns on volatility. Second, we show that the LSTM neural network architecture, which deliberately accounts for the past information through a memory state, provides a more accurate prediction of volatility compared to other machine learning methods. In particular, we show that the LSTM neural network method with only two predictors, past returns and volatility, and a finite number of lags can be used as a standard and reliable approach in the context of financial econometrics for volatility prediction.

Our results are based on a large sample of firms in the US stock market and their characteristics on a monthly frequency from 1964 to 2016. As for the predictors, we use a total of 54 firm characteristics and standard macroeconomic variables which are lagged by one period (month). The firm characteristics include also the lagged returns and volatility. The output, or the response variable, for each model is the next month realized volatility. Realized volatility is calculated as the square root of daily returns squared and summed over a month. For each machine learning method, we approximate a mapping between the predictors and the next month realized volatility. We split our data into training, validation and test samples. Instead of doing a one-time prediction, we take advantage of a recursive scheme for training and prediction. That is, each time we increase the length of the training sample by one year while rolling forward fixed-size windows of validation and test samples over time. This approach provides us with 25 subsamples and therefore a time series of performance measures such as R^2. We train and validate the selected machine learning methods over each of the 25 subsamples and therefore the set of hyperparameters for each model are varying over time. This allows the changes in the market regimes to be captured by the parameters. Finally, we use the trained models and predict realized volatility over the test samples and construct our performance measures based on the median of the performance measures across all subsamples.

Our empirical findings are threefold. First, we find that a limited number of predictors, namely, the current month realized volatility, idiosyncratic volatility, bid-ask spread and returns, respectively, account for most of the predictive power of the models. Indeed, these are the same variables that are empirically found in other studies to be most important for predicting expected returns confirming the risk-return trade-off in finance.

Second, we show that machine learning methods not only properly can capture the order and magnitude of the impact for each predictor but also the direction of these impacts, as well as the interactions between predictors without any pre-specified interaction effects in the models. In particular, our results indicate that machine learning methods can account for the stylized facts about volatility. We find that large values of the current period volatility have large and positive impact on the next period volatility, whereas, small values of the current period volatility have small and negative impact on the next period volatility. Furthermore, we find that returns have an asymmetric impact on the next period volatility. Negative returns have a higher impact on the volatility than positive returns of the same size. As for the interactions, we note that large (small) bid-ask spreads interact with large (small) values of current period volatility and have a large (small) and positive (negative) impact on the next period volatility.

Third, we show that an LSTM model, which captures the temporal dependence of predictors, can outperform the feedforward neural network and regression tree, which rely on the most recent information in the predictors.[6] In particular, our LSTM model with only volatility and return as predictors but up to one year into the past, performs as good as an

[6] We note that one can always develop a model which is based on not only the recent values of predictors but also further lags in the past. However, there are evidences as well as a common belief in the context of machine learning that an LSTM model is the best candidate for the prediction purposes where sequences of values for predictors are available. Thus, we do not develop other models (feedforward neural network and regression tree) with more than one lag of the predictors.

LSTM model with the full set of predictors and the same number of lags. One can think of our LSTM model as an alternative to the GARCH type models except that we do not need to impose any distributional assumption. However, similar to the GARCH type models we need to find the optimum number of lags through training different models across a range of lags.

Related Literature Our paper is related to several strands of the literature. We contribute to an emerging literature that investigates applications of machine learning methods in financial economics. This literature is mainly dominated by the research in empirical asset pricing (see the prominent works of Gu et al. (2020) and Chen et al. (2019) and the references therein) and also other topics such as derivatives, credit markets and computational economics.[7] Our paper is the first to document that machine learning methods, despite being notoriously complex to interpret, are not only able to accurately forecast volatility out of the sample but also to deliberately capture the properties of volatility based on a large cross-section of firms.[8]

Our paper is also related to the literature that predicts realized volatility using financial and macroeconomic variables. Realized volatility for a certain frequency (e.g., monthly) can be calculated using prices at a higher frequency (e.g., daily). The projection of this measure of volatility onto an information set results in conditional volatility. This approach is taken in French et al. (1987), Schwert (1989), Schwert and Seguin (1990), Whitelaw (1994), Marquering and Verbeek (2004). These papers use a linear model to predict the realized

[7] Hutchinson et al. (1994) and Yao et al. (2000) use deep learning methods to price and hedge derivatives. Kolm and Ritter (2019) use deep learning methods for hedging a portfolio of derivatives. As for applications in the credit market, Khandani et al. (2010) and Butaru et al. (2016) use regression trees to predict consumer credit card delinquencies and defaults whereas Sirignano et al. (2016) use deep learning to analyse mortgage prepayment and delinquency. Fuster et al. (2017) show that using machine learning for predicting borrowers' creditworthiness increases disparity in rates across race-based groups. Other applications include using neural network for portfolio selection by Heaton et al. (2016). Also, Azinovic et al. (2019) approximate recursive equilibria of economic models, and Duarte (2017) proposes an algorithm for solving a large class of nonlinear continuous-time models in finance and economics. Bashchenko and Marchal (2019) use deep learning to detect jumps in financial time series.

[8] Rossi (2018) uses a regression tree model to predict volatility of S&P 500 index using twelve macroeconomic predictors and finds an out-of-sample R^2 of about 40%. He finds time-varying and economically valuable optimal portfolio weights for a representative mean-variance investor based on the forecasts of expected returns and volatility.

volatility of the market index using a limited number of macroeconomic variables and find very low out-of-sample prediction power.[9] Ludvigsona and Ng (2007) and Christiansen et al. (2012) also use the same approach but with larger sets of predictors. In contrast to the previous works which use only a few predictors to estimate the conditional volatility. Ludvigsona and Ng (2007) in their study of the risk-return relation, use dimension reduction techniques to summarize 209 economic indicators and 172 financial indicators into only a few factors, and then estimate a linear regression of the realized volatility on the estimated factors. They achieve an in-sample R^2 of about 40% but they do not report an out-of-sample R^2. Instead, they show that the ratio of the out-of-sample forecast error of their model compared to a simple AR(1) model of realized volatility is as low as 0.7. Similar to their work we also find that there are only a few predictors driving most of the prediction power for the next period volatility. Christiansen et al. (2012) use a Bayesian model averaging approach for predicting realized volatility of several asset classes based on 38 financial and economic variables. For the case of equity volatility, they find an out-of-sample R^2 ranging from -2% to 7.8%. However, unlike these two works and instead of the equity market index, we use the firms in the US stock markets for which the 46 firm characteristics exist every month from 1964 to 2016. We find an out-of-sample R^2 ranging from 61% for our linear model to 83% for our neural network model. Finally and most importantly, we are also able to capture the stylized facts about volatility.

The rest of the paper is organized as follows. Section 2 describes our methodology which includes a description of machine learning methods. Section 3 presents the empirical study. This includes a description of our data, the model performance and interpretation of the findings. Section 4 concludes.

[9] Engle and Rangel (2008) and Engle et al. (2007) use Spline-GARCH and GARCH-MIDAS models to study the economic determinants of stock market volatility.

2 Methodology

In this section, we explain the machine learning models that we use in our empirical study. We focus on three classes of models that cover a wide range of methods. The first model is an elastic net, which belongs to the class of linear regression models. An elastic net model is a convex combination of lasso and ridge regression models, with the plain vanilla regression model being a special case of it. Such models are linear and simple to understand and their performance can be compared against the nonlinear models. Second, we use regression tree models that belong to the class of ensemble methods. These models are nonlinear while still tractable and easy to understand. Third, we use neural network models, which are nonlinear and the least tractable. The three models together provide us valuable information about the trade-off between simplicity and model performance. In addition, we can compare the model performance during boom and bust cycles where the nonlinearities may become more pronounced. We follow Gu et al. (2020) in our choice of the predictive model and sketch the general form of our model as follows. A realized stock return volatility could be described as an additive prediction error model of the form:

$$\sigma_{i,t} = E_{t-1}[\sigma_{i,t}] + \epsilon_{i,t} \tag{1}$$

where

$$E_{t-1}[\sigma_{i,t}] = f(x_{i,t-1}). \tag{2}$$

Here stocks are indexed by $i = 1, ..., N_t$, where N_t is the number of stocks available at time $t = 0, 1, ..., T-1$, and x denotes a P-dimensional vector of predictors. The conditional expectation in (2) is a flexible function f of these predictors, which does not change across stocks and over time. While this is a restriction, it leads to a stable estimate of realized volatility as it allows the model to exploit information from the entire panel. Our objective is to find an approximation of the function f that maximizes the out-of-sample explanatory power for realized stock return volatility.

Our analysis is based on monthly data. We follow French et al. (1987) and Schwert (1989) to obtain a measure of monthly volatility for returns. In particular, we use the time series variation of daily returns to construct monthly realised volatility as follows:

$$\sigma_{i,t} = \sqrt{\sum_{d=1}^{n_t} r^2_{d,i,t}} \tag{3}$$

where n_t is the number of trading days in the t-th month, and $r_{d,t,i}$ indicates the daily excess return of the stock i on the d-th trading day of the t-th month.

2.1 Sample Splitting and Regularization

It is well-known that machine learning methods are prone to overfit the data. Regularization is a way to avoid overfitting and the extent to which a model is punished by regularization is governed by hyperparameters. For most models, the regularization and choice of hyperparameters is not a standard procedure and can be an extensive task. In particular, the choice of hyperparameters requires one to set aside part of the training data as a validation data. The validation step can be thought as simulating the out-of-sample performance of the model. That is, we look for hyperparameters that minimize the forecast errors over the validation sample for a model that is estimated over the training data. Overall, the process of estimation and prediction, which entails tuning hyperparameters and regularization, requires one to partition the sample into three parts: training, validation and test sample.

One simple way to do this partitioning is to split the sample into fixed periods of training, validation and test. This method requires doing the training, validation and test once and thus is the least computationally expensive splitting method. Although simple, the measure of model performance obtained from this method might not accurately reflect regime changes in the data over time. This weakness can be addressed by using a rolling method in which the training, validation and test windows rollover time. That is, by shifting the windows forward the most recent data points come into play whereas the oldest data point are excluded from the windows. However, the exclusion of the oldest information from the modelling might have a significant impact on the model performance. An example is when the rolling estimation window is leaving a crisis period behind. Another method is a recursive approach where the window expands over the sample period and thus the model takes the old information into account. It is obvious that the recursive approach is computationally expensive. In this paper, we use the method of rolling windows and explain it in more detail in Section 3.1.[10]

2.2 Linear Models

In a linear model, function f in (2) can be represented by a linear function of predictors and parameters:

$$f(x_{i,t}) = x'_{i,t}\beta. \tag{4}$$

The objective function can be written as

$$\mathcal{L}(\beta;\cdot) = \mathcal{L}(\beta) + \pi(\beta;\cdot) \tag{5}$$

where we consider three variants for function $\mathcal{L}(\beta)$ namely least square error (ℓ_2) and least absolute error (ℓ_1),

$$\ell_2 = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}(\sigma_{i,t} - f(x_{i,t-1}))^2 \tag{6}$$

$$\ell_1 = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}|\sigma_{i,t} - f(x_{i,t-1})| \tag{7}$$

and the Huber loss function (ℓ_H), which is robust to outliers, and is a combination of the ℓ_1 measure (for large errors) and the ℓ_2 measure (for small errors). The contribution of each

[10] Our findings are robust to the choice of sample splitting approach. We obtain our results, described in Section 3, also under the third approach, that is the recursive approach, and find almost no change in our main results. The average change in our performance measure R^2 is less than 1%.

measure is controlled by parameter δ which is usually a certain quantile of absolute errors:

$$\ell_H = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t} H(\sigma_{i,t} - f(x_{i,t-1}), \delta) \tag{8}$$

where

$$H(\epsilon, \delta) = \begin{cases} \epsilon^2 & |\epsilon| \leq \delta \\ 2\delta|\epsilon| - \delta^2 & |\epsilon| > \delta \end{cases}$$

The second term in the objective function (5) is the elastic-net penalty function:

$$\pi(\beta; \alpha_1, \alpha_2) = \alpha_1\Big(\alpha_2\sum_{j=1}^{P}|\beta_j| + \frac{1}{2}(1-\alpha_2)\sum_{j=1}^{P}\beta_j^2\Big) \tag{9}$$

This function allows for the penalization of β parameters with the two hyperparameters α_1 and α_2 controlling the extent of penalization in different ways. With $\alpha_2 = 0$ we would have ℓ_2 parameter penalization which avoids large estimated β parameters by shrinking them to be close to zero. So in this case we would have a ridge regression. On the other hand, $\alpha_2 = 1$ corresponds to the lasso regression which uses ℓ_1 parameter penalization and works as a variable selection method by eliminating a subset of predictors and setting their parameters to zero. Thus, the hyperparameters α_2 with a value between 0 and 1 provides both selection and shrinkage penalty functions for the parameters of the model. Finally, the hyperparameters α_1 controls the extent of parameter penalization in the objective function.

2.3 Tree-Based Models

One caveat of linear models is that they do not account for the potential nonlinearities of the response function. Adding interactions into the linear model might be computationally infeasible when the number of predictors is large and one needs to have prior knowledge about a subset of interactive terms. Tree-based models provide the opportunity to investigate such nonlinearities in a relatively easy way. Tree-based models are non-parametric and based on the idea of partitioning the predictors' space into several distinct regions and approximating the response function by taking the average values of the outcome variable for the predictors within each region. In this process which is known as the recursive binary splitting approach the predictor X_j and splitting point s are selected such that the resulting regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ lead to the greatest reduction in the loss function. Once the two regions are found, the algorithm further splits one of the two regions which gives more depth to the tree. This process continues until a stopping criterion is reached. A function f can be approximated by a tree with depth D and 2^D terminal nodes as:

$$f(x_{i,t}; \beta, D) = \sum_{c=1}^{2^D} \beta_c \mathbb{1}_{[x_{i,t} \in R_{c,D}]} \tag{10}$$

where $R_{c,D}$ is the region c at depth D, and β_c is the sample average of responses in this region. Despite their simple yet promising structure, tree-based models are prone to overfit the data and thus need to be regularized.

Our focus here is on the Gradient Boosting Regression Tree (GBRT) algorithm introduced by Friedman et al. (2000) and Friedman (2001), which is a powerful method to regularize and enhance the predictive power of the regression trees.[11] Instead of growing a single deep tree that is prone to overfitting, the idea in boosting is to sequentially grow many shallow trees and combine their forecasts. That is, in each step a new weak learner (tree) compensates for the shortcomings of the current weak learners. These shortcomings are the residuals from the fitted trees and can be identified by the negative gradients. This sequential approach allows the learning to happen slowly from many weak learners by growing each tree on the residuals from the previous tree. There are several hyperparameter that control the extent of regularization: the number of trees $\mathcal{T}$, the shrinkage parameter λ which controls the contribution of each tree and finally the depth of each tree as D. Our function f can be

[11] Caruana and Niculescu-Mizil (2006) show that results obtained from boosting methods often dominate those of random forest and bagging.

approximated as:

$$f(x_{i,t}; \mathcal{T}, \lambda, D) = \sum_{\tau=1}^{\mathcal{T}} \lambda f_\tau(x_{i,t}; \beta, D) \tag{11}$$

with the objective function being either of the functions (6)-(8).

2.4 Neural Network Models

An alternative machine learning method that accounts for nonlinearities is the class of artificial neural network models. The nonlinearities inherent in the neural network models are due to representations that are expressed in terms of other, simpler representations. Despite their success, neural network models are complex and difficult to interpret.

2.4.1 Feedforward Neural Networks

We first use a simple form of neural network known as multilayer perceptron or feedforward deep network (FF). In a feedforward network, the relationship between the input layer of predictors and the response variable is made by one or more hidden layers that interact with each other and nonlinearly transform the predictors. This nonlinear transformation is based on an element-wise operating activation function. We choose a popular activation function known as the Rectified Linear Unit (ReLU):

$$\mathcal{A}_{relu}(x) = max(x, 0) \tag{12}$$

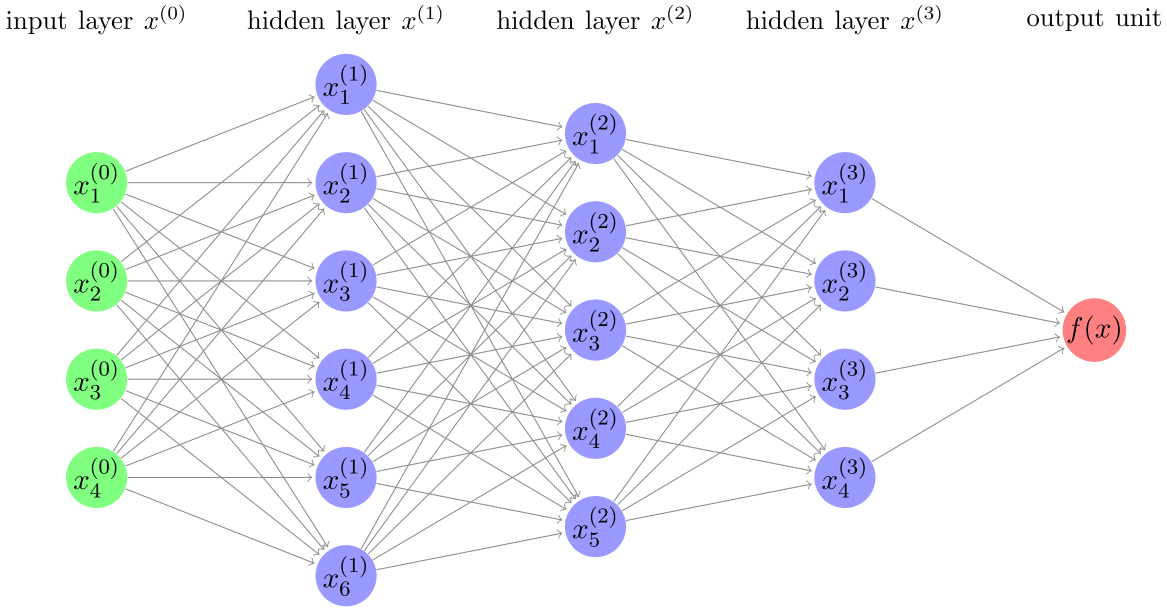
The ReLU function results in the l-th hidden layer once applied to the $(l-1)$-th layer:

$$x^{(l)} = \mathcal{A}_{relu}\Big(\beta_0^{(l-1)} + \sum_{u=1}^{U^{(l-1)}} \beta_u^{(l-1)} x_u^{(l-1)}\Big) \tag{13}$$

where $x^{(l)} = (x_1^{(l)}, \ldots, x_{U^{(l)}}^{(l)}) \in \mathrm{I\!R}^{\mathrm{U}^{(l)}}$ is the vector of outputs for layer l, where $l = 1, \ldots, L$, and $x_u^{(l)}$ is the output of unit u in layer l, where $u = 1, \ldots, U^{(l)}$, and $U^{(l)}$ denotes the number of units in layer l. For $l = 0$, vector $x^{(0)} = (x_1^{(0)}, \ldots, x_U^{(0)})$ denotes the input layer which contains the predictor variables. Schema 1 illustrates an example of a network with an input layer (in green), three hidden layers (in blue) and the output unit (in red). Each unit in the first hidden layer receives a linearly weighted signal from input predictors and applies a nonlinear transformation on it, as in (13), before sending its output the units in the next hidden layer. Finally, the information in the last hidden layer is linearly aggregated into a final forecast with the following functional form:

$$f(x; \beta, U^{(l)}, L) = \beta_0^{(L)} + \sum_{u=1}^{U^{(L)}} \beta_u^{(L)} x_u^{(L)}.$$

Schema 1: Feedforward Neural Network with Three Hidden Layers



Estimation of a network entails finding the weights that best make the relationships between input and output variables. For large networks, the number of such weights could be significantly large which makes the estimation computationally intensive. To put this into perspective, the network structure shown in Schema 1 has a total of $94 = (1+4) \times 6 + (1+6) \times 5 + (1+5) \times 4 + 5$ parameters to be estimated. In general, there are $(1 + U^{(l-1)}) \times U^{(l)}$ parameters in each hidden layer l and $1 + U^{(L)}$ parameters in the output layer.

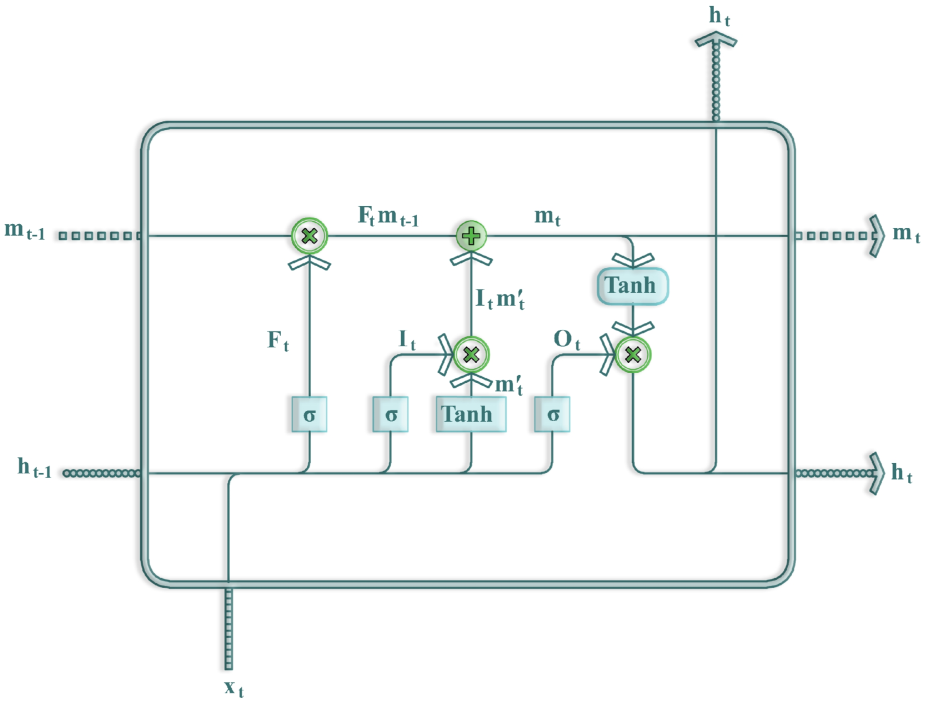
To estimate the parameters we minimize the loss function using the stochastic gradient descent (SGD) approach. SGD differs from gradient descent optimization in that it does not use the entire sample to minimize the loss. Instead, it calculates the gradients using a small and randomly selected subset of samples. The difference between the size of the subsamples used in SGD and the entire sample size, calls for a certain number of iterations at each step of the optimization, so that all data points are contributing to the estimation. The smaller is the size of the subsample, the faster is the algorithm and more iterations are needed at each step. However, the faster computations of gradients due to the small sample size comes at the cost of lower accuracy.

We tune the initial learning rate (or descent step size) in the SGD and let the tuned rate be adaptively shrunken towards zero as the gradient approaches zero. We also use early stopping to terminate training when the validation score is not improving by at least the pre-specified tolerance. Finally, we repeat the whole training and forecasting procedure several times and calculate our final forecast using the average of the forecasts from several networks. Although these networks have the same specifications, their final forecasts are different due to their different random seeds used for initialization.

2.4.2 Long Short-Term Memory Neural Networks

Our specification in the additive prediction error model (1)-(2) captures the relation between the predictors in the current period and the next period volatility. None of the models we have discussed so far accounts for temporal dependencies beyond the first lag. However, stock return volatilities are known to be persistent over time and thus it makes sense to examine if further lags of volatilities and other predictors provide a better estimate and prediction of future realized volatilities. Thus, in (1)-(2) we are interested in a model that allows for a more general function $f(x_{i,t-k}), k = 0, 1, \dots, K$ where K is the number of predictors' lags in the model. The so-called Recurrent Neural Networks (RNN) capture such dependencies through a memory structure. However, the standard RNN may not perform well if it needs to carry some information from long in the past because the memory that contains such information fades away and thus it may not contribute to the outcome. A special case of RNN network called LSTM network, proposed by Hochreiter and Schmidhuber (1997), resolved this issue by embedding a mechanism for keeping or throwing away information from the past. LSTM models are widely used and proved to be very successful for a variety of problems in different domains.[12] Similar to the RNN network, the units in a LSTM network are of the form of repeating modules except that they are more sophisticated. This is shown more precisely in Schema 2 which depicts the memory cell. The neural networks embedded inside the cell can maintain or drop information over time by controlling the information flow into and out of the cell.

Schema 2: Schematic of a Unit in the LSTM Neural Network



The key predictor of the LSTM architecture is the cell state or the memory represented by the straight line stretched from m_{t-1} to m_t. Past information flows through the cell state to the future but there is a chance that it will be altered on the way out by three gates. These

[12] LSTM-based systems have caught the attention of both academic and industry and are heavily in-use by giant technology companies. Such systems can learn to translate languages, control robots, analyse images, summarise documents, recognise speech, videos and handwriting among many other applications.

gates are neural networks with logistic activation functions that result in values ranging from 0 to 1 and assign a value to each instance in the memory. For instance, the *forget* gate with an output value close to 0 erases most of the past memory. The *input* gate with an output value close to 1 allows the current information to highly influence the past memory.[13] Finally, the *output* gate with some value between 0 and 1 decides how much of this already altered memory to contribute to the current time prediction. A copy of the prediction, as well as the long-term memory, will be kept to contribute to the information in the next time step.

We formalize this mechanism in the following. At each time, the long term memory can be influenced by the long term memory carried over from previous period, m_{t-1}, and the current information, m'_t. The extent to which each of these two components can contribute to m_t is regulated by the input gate I_t and forget gate F_t,

$$m_t = F_t \otimes m_{t-1} + I_t \otimes m'_t \tag{14}$$

where $\otimes$ is point-wise multiplication. The long term memory m_t then will be used in two ways. It will be carried over to influence the next period memory m_{t+1} in the same way as in (14), and it also results to the prediction for the current cell,

$$h_t = O_t \otimes \mathcal{A}_{tahn}(m_t) \tag{15}$$

where $\mathcal{A}_{tahn}$ is the hyperbolic tangent activation function and O_t works in a similar way as F_t and I_t and decides about the influence of the memory to the output of the current cell. It remains to define the functional form of the gates F_t, I_t, O_t and the potential new candidate

[13] In fact the input gate adds up to the cell state by deciding which of the possible new candidates should be used. These new candidates are chosen by a *tahn* activation function.

m'_t,

$$
\begin{aligned}
F_t &= \mathcal{A}_{log}(\beta_h^f h_{t-1} + \beta_x^f x_t + b^f) \\
I_t &= \mathcal{A}_{log}(\beta_h^i h_{t-1} + \beta_x^i x_t + b^i) \\
O_t &= \mathcal{A}_{log}(\beta_h^o h_{t-1} + \beta_x^o x_t + b^o) \\
m'_t &= \mathcal{A}_{tahn}(\beta_h^m h_{t-1} + \beta_x^m x_t + b^m)
\end{aligned}
$$

where $\mathcal{A}_{log}$ is the logistic activation functions, h_{t-1} is the vector of predictions from previous period, x_t is the current period vector of input data, and $\{\beta_h^j, \beta_x^j\}$ and $\{b^j\}$ for $j = f, i, o, m$ are the matrices of weights and vector of biases, respectively.

Although LSTM architecture has proved to be very successful in several domains it entails estimating a larger set of parameters compared to a standard feedforward neural network. The number of parameters for layer l of the LSTM network can be calculated as $4 \times U^{(l)} \times (1 + U^{(l)} + U^{(l-1)})$ where $U^{(l)}$ denotes the number of units in layer l. Table 1 reports the number of parameters for various network architectures. In particular, we report the changes in the number of parameters for feedforward and LSTM networks across changes in the (1) number of predictors, (2) number of units in each layer, and (3) number of layers that we use in our empirical analysis. We note that in both models the number of parameters significantly increases by changing the number of predictors from 2 to 46 and 54 (columns I to III for FF, and columns IV to VI for LSTM). Adding one more layer to the networks substantially increases the number of parameters with magnitudes depending on the number of input predictors. On average, the LSTM network has about 5 times more parameters compared to the feedforward network.

[Insert Table 1 here]

2.5 Model Performance and Predictor's Importance

We use Root Mean Squared Error (RMSE), Mean Squared Relative Error (MSRE), Mean Absolute Relative Error (MARE) and R^2 as our performance measures for out-of-sample forecast evaluations. As we explained in Section 2.1 we split the sample into three disjoint sets of training, validation and testing. We train the model over the training period, tune the hyperparameters over the validation period, and calculate the performance measures for the predictions over the test sample year y ($y = 0, 1, \ldots, 24$):

$$RMSE_y = \sqrt{\frac{1}{12}\sum_{t=12y}^{12y+11}\sum_{i=1}^{N_t}\left(\sigma_{i,t+1} - \hat{\sigma}_{i,t+1}\right)^2}$$

$$MSRE_y = \frac{1}{12}\sum_{t=12y}^{12y+11}\sum_{i=1}^{N_t}\left(\frac{\sigma_{i,t+1} - \hat{\sigma}_{i,t+1}}{\sigma_{i,t+1}}\right)^2$$

$$MARE_y = \frac{1}{12}\sum_{t=12y}^{12y+11}\sum_{i=1}^{N_t}\left|\frac{\sigma_{i,t+1} - \hat{\sigma}_{i,t+1}}{\sigma_{i,t+1}}\right|$$

$$R_y^2 = 1 - \frac{\sum\limits_{t=12y}^{12y+11}\sum\limits_{i=1}^{N_t}(\sigma_{i,t+1} - \hat{\sigma}_{i,t+1})^2}{\sum\limits_{t=12y}^{12y+11}\sum\limits_{i=1}^{N_t}(\sigma_{i,t+1} - \bar{\sigma})^2}$$

where σ is the annualized realized volatility and $\hat{\sigma}$ is the annualized realized volatility predicted by our models.

In addition to calculating the performance measures for each model we are interested to see the contribution of each predictor to the performance of each model and eventually arrive at a shortlist of predictors that drive most of the performance for each model. One way to achieve this goal is to check if there will be a decrease in the performance of the model when a particular predictor is absent from the model or replaced by random noise. Thus, we implement permutation importance for each predictor of a model. More precisely, after training the model, we shuffle the values of each predictor in the test sample and check if there is a decrease in the R^2 of the model. We repeat this process for each predictor separately and sort them by the reduction they cause to the forecast power of the model. We list predictors with the highest reduction in the forecasting power as the most important predictors. As an alternative approach, we also use the recent advances in machine learning techniques that try to make the results of machine learning models more explainable. In particular, we use the methodologies developed by Lundberg and Lee (2017) and Lundberg et al. (2020) to evaluate the direction of the impact of predictors as well as the interactions among them and their final impact on the response variable. We provide a detailed explanation for this method in Appendix B.

3 Empirical Study

We start this section by describing the data, and then discussing the results of the three models namely the elastic net model, the gradient boosting regression tree model and the feedforward neural network model. Then we discuss which predictors are the main drivers of these results and how they contributed to the prediction. We conclude the section by discussing the results of the LSTM model that accounts for the temporal dependencies in the data.

3.1 Data

Our data consist of 53 years of observations for all securities on CRSP/Compustat and CRSP databases from January 1964 to December 2016. These are 46 monthly and annual firm-specific characteristics constructed either from accounting variables available at the CRSP/Compustat database or from past returns available at CRSP. CRSP contains price data for about 31000 firms. But this number reduces to around 10000 stocks when filtered for firms with all firm information available at each month.[14] We follow the existing conventions regarding the frequency and usage of variables. The firm characteristics which are available in monthly frequency are updated at the end of each month for use in the next month whereas yearly updated variables are updated at the end of each June. That is, data from the fiscal year ending in calendar year $y-1$ are used for estimation starting in June of year y until May of year $y+1$. So, for instance, the data available from June 2015 to May 2016 are due to the firm conditions during the fiscal year 2014. This will have an implication for

[14] Chen et al. (2019), Kelly et al. (2019) and Freyberger et al. (2020) also use a similar dataset.

modelling purposes. In our case, the implication for training (and validation) of the model is that when the model is learning the relation between predictors at month t and the response variable at month $t+1$, depending on being in the first or second half of the calendar year, the model does not have access to the firms' information exactly at time t. For instance, when we train the model over the calendar year 2016, during the first half of the year, the model is learning the relation between the response variable at month t and the (annually updated) predictors at the fiscal year 2014. Accordingly, during the second half of the year, the model is learning the relation between the response variable at month t and the (annually updated) predictors at the fiscal year 2015. In contrast, for the case of monthly updated predictors which are available at the end of each month, the model is learning the relation between the response variable at month $t+1$ and predictors at month t.

We also use 8 standard macroeconomic predictors following Welch and Goyal (2008), including dividend-price ratio, earnings-price ratio, book-to-market ratio, net equity expansion, Treasury-bill rate, term spread, default spread, and stock variance. In Appendix A we show the list and definition of all variables. Overall, we have 54 firm characteristics and macroeconomic predictors, 30 of which are in annual frequency and the rest are in monthly frequency. This also includes the lag of the realised volatility. Throughout the paper, we use the term volatility instead of realised volatility. According to the specification in (2), all features are lagged by one period to predict the next period volatility.

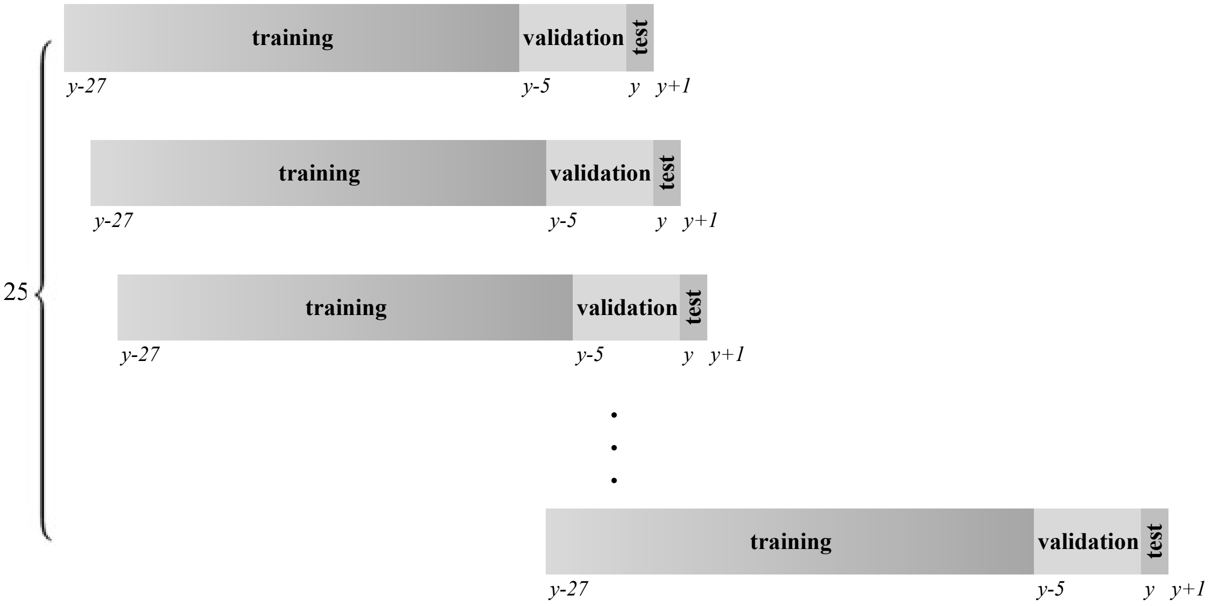
To limit the effect of possibly spurious outliers we drop the 1% smallest and largest values of the data. We further perform several transformation schemes to account for outliers and deal with differences in the scales of predictors. In particular, we perform a real-time standardization of the data by subtracting the mean and scaling by the standard deviation. We apply this transformation to each subsample separately. That is, we standardize the training sample and then use the estimated mean and standard deviation to standardize the validation and test samples. In an unreported robustness analysis, we also apply two other transformation procedures. First, we normalize the data to the range between 0 and 1. Second, for each month we transform the values into quantiles.[15] We do this for each month

[15] See Kozak et al. (2020) for more details on this approach.

by cross-sectionally ranking the values across predictors and then dividing the ranks by the number of observations for that month. We recalculate all our results discussed in Section 3.2 with the data transformed by these two latter schemes and find no significant changes in the results.

As shown in Schema 3 we divide the 53 years of data into 22 years of the training data, 5 years of validation data, and the remaining 25 years for out-of-sample testing. This results in 25 subsamples where the training, validation and test periods have fixed sizes of 22 years, 5 years and 1 year, respectively, and at each time we roll them over by 1 year. For instance, the first subsample consists of a training period 1964-1986, validation period 1987-1991, and the test year 1992. Similarly, the last subsample consists of a training period of 1988-2010, validation period 2011-2015, and the test year 2016. As a result, our predictions for each month in 2016 are based on the model that we train over the period 1988 to 2010. In an unreported robustness analysis, we find no changes in our main results if we use an expanding window for the training sample instead of a fixed size window. In this case, the last subsample consists of a training period of 1964-2010, validation period 2011-2015, and the test year 2016.

Schema 3: Sample Split Scheme



The prediction performance measures for each model in the next section is based on the median of the performance measures across all the 25 samples present in this figure.

3.2 Model Performance

In this section, we present the results of the selected methods for predicting stock return volatility. Table 2 shows the out-of-sample RMSE, MSRE, MARE and R^2 for the ELN, GBRT and FF neural network model with one to four hidden layers ($\text{FF}_l, l = 1, ..., 4$). All three types of models include the full set of predictors. In the case of the FF neural network model, FF_1 represents the model with only one hidden layer of 16 units, FF_2 represents the model with two hidden layers of 16 and 8 units, FF_3 represents the model with three hidden layers of 16, 8 and 4 units, and finally, FF_4 represents the model with four hidden layers of 16, 8, 4 and 2 units. Numbers in the table are obtained as the median of the measures over the 25 subsamples described in Sections 2.1 and 3.1. All performance measures change significantly when we go from the ELN model to the nonlinear models. This can be an indication of the inherent nonlinearities leading to changes in the volatilities of stock return. Among the nonlinear models, the gradient boosted regression tree performs almost as good as the feedforward neural networks. The best performance is obtained using the FF_4 model.

[Insert Table 2 here]

While Table 2 reports the medians of the performance measures across subsamples, we can also observe the evolution of these measures across subsamples. Figure 1 shows the evolution of these measures for all of the three models. The subplots show that the two nonlinear models closely follow each other and perform far better than the linear model. The only exception is during the financial crisis in 2008 where the FF neural network performs worse.

[Insert Figure 1 here]

In Figure 2 we reconstruct the time series of predicted volatilities in the test sample against the actual volatilities from 1992 to 2016. For each year in the test sample, we have monthly predicted volatilities for the stocks that are available in that month. For each month, we average the predicted volatilities across those stocks. This results in an out-of-sample time series of monthly volatilities. We repeat the same procedure for the actual volatilities. Our out-of-sample prediction of volatility is on average 44.1% for the FF model, and 44.4% for the GBRT model, against the actual volatility of 43.8% for the stocks in our sample. The FF model has four hidden layers and all predictors are present in both models.

[Insert Figure 2 here]

3.3 Predictors That Matter Most

In this section, we further investigate the contribution of each of the predictors to the changes in the dynamics of the performance measures in Figure 1.

3.3.1 Order of Importance

The most important predictor across all models and all the subsamples is the lagged volatility which by far dominates all other predictors. We intentionally omit this predictor from the list of predictors in Figure 3 and also from the plots in the upper side of Figures 4 to 6 to provide a readable visualisation of the importance of other predictors. However, we discuss the importance of volatility together with other main predictors in all other subsequent figures.

[Insert Figure 3 here]

Figure 3 reveals the importance of each predictor for all models but sorted only for the FF neural network model. Putting aside the volatility as the best predictor, the idiosyncratic volatility (*idiovol*), bid-ask spread (*spread*) and returns (*ret*) are the most important predictors of the next period volatility for both the feedforward neural network and GBRT models.[16] However, there are some differences between them. For instance, according to the GBRT, much of the prediction power is due to the idiosyncratic volatility compared to the bid-ask spread and returns. Finally, the linear regression model in the last column highlights the bid-ask spread as the best predictor followed by the closeness to past year high (*real2high*). In the next three figures, we investigate each model individually.

Figures 4–6 show the importance of all predictors (excluding volatility) in the upper panel, and the main four predictors (including volatility) in the lower panel across subsamples. Each year on the horizontal axis represents the test period for the corresponding subsample. For instance, the forecasts for the year 2016 are obtained by the model trained over the sample from 1988 to 2010 and validated over the sample from 2011 to 2015. The plot in the lower panel of Figure 4 shows that the importance of the volatility has slightly deteriorated over time whereas the bid-ask spread, returns and idiosyncratic volatility have gained more importance over time. The plot in the upper panel also reveals that, apart from the main predictors, other predictors such as market capitalization (*lme*), total assets (*at*), turnover (*lturnover*), closeness to past year high (*real2high*) contribute in predicting the next period volatility. An interesting observation is that in 2009 the default spread (*dfy*) shows up and makes the spread and returns less important.

[Insert Figures 4–6 here]

Figure 5 belongs to the GBRT model. The plot in the lower panel shows almost the same dynamics and order for the main predictors in the GBRT model. But visible differences between the GBRT and FF neural network models become evident in the plot in the upper panel of Figure 5. That is, unlike the FF neural network model, other predictors remain almost silent over time with the exception being the market capitalization (*lme*). This means that the complexity of the FF neural network allows the model to involve more predictors in the prediction.

Finally, Figure 6 shows that for the ELN model, volatility drives much of the predictive power for predicting the next period volatility across all the times. The second driver is

[16] Ludvigsona and Ng (2007) also find that volatility can be explained by only a few factors.

the bid-ask spread. But it tends to become weak during the stress times 1998-99, 2008-09 and 2015-16. During these times the idiosyncratic volatility shows up and contribute to the prediction. On the other hand, the linear model fails to highlight returns as an important predictor.

These results suggest that one should judge the overall success of different models for forecasting volatility with caution and further examine the performance of the models over different market regimes. All of the figures in this section have been based on the aggregate importance of the predictors or their dynamics over subsamples. For the rest of the Section 3.3, we derive our results using only the last subsample with the test period of 2016. Our main results remain unchanged if we use any of the subsamples.

3.3.2 Interpretation: Directions and Interactions

A natural step in modelling data is to study not only the magnitude of the impact but also the directions and possible interactions of predictors with respect to the response variable. However, a trade-off exists between the complexity and interpretability of models. For instance, one would achieve a high degree of interpretability by using a linear regression framework at the cost of the incapability in capturing potential nonlinearities. On the other hand, more complex models such as neural networks offer high potentials in capturing nonlinearities and interactions at the cost of being barely interpretable. As a result, the ongoing research in making complex models more interpretable has received significant attention along with the developments in the models. In certain domains such as financial economics, one would easily forgo the complexity (and sometimes predictive power) of the model in favour of interpretability. We devote this section to this issue and rely on the most recent advances in interpreting the models of the previous section that showed to be successful in predicting realized volatilities.[17]

[17] We use the methodologies developed by Lundberg and Lee (2017) and Lundberg et al. (2020) to evaluate the direction of the impact of predictors as well as the interactions among them and their final impact on the output of the model. All figures in this subsection are derived based on these methods. We provide a detailed explanation for these methods in Appendix B.

Before investigating the direction and interaction of predictors, we list the top ten predictors for each model in Figure 7. According to this figure, there are other important predictors (*tbl*, *lme*, *st_rev*, r_{12_2}, *at*) which are common across the models. It is interesting to note that the interest rate (*tbl*) appeared among the top five predictors. This is in line with the findings of, for example, Breen et al. (1989) who show that knowledge of the one-month interest rate is useful in forecasting the sign as well as the variance of the excess return on stocks. Glosten et al. (1993) and Engle and Patton (2001) find similar results.

[Insert Figure 7 here]

Figure 8 shows the distribution of the impact of the top four predictors on the next period volatility. It shows the predictors on the vertical axis sorted by their importance. It also shows the magnitude and direction of the impact of each predictor on the next period volatility in the horizontal axis. For each predictor, the colour changes depending on its value from low (red) to high (green).

[Insert Figure 8 here]

This figure reveals several important predictors of stock return volatility: (1) volatility in the first row is the most important predictor for the next period volatility. It indicates that large values (green dots) of the current period volatility have large and positive impact on the next period volatility, whereas, small values (red dots) of the current period volatility have small and negative impact on the next period volatility and finally, middle-sized values (yellow dots) of the current period volatility have very small and mixed impact on the next period volatility. (2) returns of either sign increase the next period volatility, whereas middle-sized values (i.e. almost no change in the stock price from one period to the next) of the returns have almost no impact on the next period volatility. This observation indicates that large changes of either sign in the current period returns are going to be followed by an increase in the next period volatility.[18] Observations (1) and (2) highlight some stylized

[18] Due to the overlaps of the colours in the plot, it is not possible to compare the magnitude of the impact by positive and negative returns. However, the so-called news impact curve indicates that returns often have an asymmetric effect on the volatility meaning that negative returns will have a larger impact on the volatility compared to the positive returns of the same size.

facts about the volatilities of stock return. The so-called volatility clustering first observed by Mandelbrot (1963) indicates that large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes. Also absolute or squared returns display a positive, significant and slowly decaying autocorrelation function. We also note from Figure 8 that the linear model does not picture this phenomenon as it fails to identify returns as an important predictor.[19] (3) idiosyncratic volatility has the same impact as volatility on the next period volatility, but it goes in the opposite direction. That is, large (small) values of the idiosyncratic volatility have a large (small) and negative (positive) impact on the next period volatility. Finally, (4) bid-ask spread has a similar impact as volatility on the next period volatility except that its magnitude is smaller.

[Insert Figures 9–10 here]

Figures 9 and 10 are similar to Figure 8 except that the colours on the vertical axis represent the values of a predictor which has interacted most with the main predictor on the horizontal axis. Both FF and GBRT models indicate volatility as the most interacting predictor with spread. We note that large (small) values of spread interact with large (small) values of volatility and have a large (small) and positive (negative) impact on the next period volatility. Another interesting observation is the asymmetric impact of returns on the next period volatility. As expected, negative returns have a higher impact on the volatility than positive returns of the same size. Results from these figures are important as they indicate that each predictor serves multiple roles in predicting the output of the model.

Our results so far indicate that (1) nonlinear models are superior in predicting volatility and (2) they can capture important stylized facts about stock return volatility, which are the volatility clustering and the asymmetric impact of returns. Finally (3) most of the prediction power is due to only a few predictors. Our objective in the next section is to build on these findings and investigate the LSTM model which can more accurately account for these properties.

[19] We note that one can always insert the negative and positive returns as different predictors in a linear model and capture the impact of returns.

3.4 Does Temporal Dependence Matter?

The nonlinear models in the previous sections showed to have the potential to predict the next period volatility using few predictors. In particular, these models were able to highlight the volatility clustering and asymmetric impact of returns on volatility. Similarly, over the past several decades, GARCH type models proved to be very successful in capturing stylized facts about stock return volatility in the context of financial econometrics. The next period variance in these models can be modelled as an additive function of past variances and past squared innovations or returns. Although one can estimate the model for many lags of variances and squared returns, empirical observations reveal that it is often enough to use the model with only the first lags of variance and squared return. In the context of machine learning methods and for volatility prediction, one can think of a neural network model as a more generic mapping between the current period values of predictors and the next period volatility. In this section, we further extend such mapping by going beyond the models with only one lag of predictors and see whether the lags of the predictors further in the past could improve the prediction. As explained in detail in Section 2.4.2, we use an LSTM model for this purpose. An LSTM model can deliberately account for the information further in the past through its memory cell and predict the future volatility.

We first explore the ability of the model in accounting for the time dependencies by going beyond the first lag of predictors and including more lags in the model. This allows us to examine the extent to which past information is relevant for future volatilities. We decide to go up to 11 months into past and thus estimate 11 different models. For instance, the last model with 11 lags for each predictor uses the information up to January 2015 to predict the volatility of January 2016.[20] We then investigate whether a simple model with few predictors but a rich lag specification can perform as good as a model with a full set of predictors. We develop three versions of the LSTM model as follows. Similar to the GARCH(p,q) model, we start with a simple model where only the volatilities and returns from previous periods can explain the next period volatility. In the second step, we let also past firm-specific predictors

[20] Our test period consists of 12 months and thus we cap the lag window by 11 months to keep the LSTM model comparable with the previous models discussed in Section 3.2.

contribute to the prediction of future volatility.[21] Finally, we train an LSTM model where not only past volatilities, returns and firm characteristics but also past macroeconomic predictors contribute to the prediction of volatility. This specification allows us to study the potential incremental improvement in the prediction due to a different set of predictors.

All together we perform the training, validation and prediction steps $3 \times 11 \times 4 \times 25$ times for models with 3 different sets of predictors, 11 different lag structures, 4 different layer specifications and finally over 25 subsamples. Figure 11 shows our out-of-sample performance measures, aggregated over 25 subsamples, for 3×11 models with 4 hidden layers.

[Insert Figures 11 here]

This figure reveals several interesting patterns for the performance measures on the vertical axis and across lags on the horizontal axis. We base our interpretation on the plot for R^2. The performance of the LSTM models with the full set of firm characteristics ($model_{r,v,\tilde{x}}$) and models with the full set of firm characteristics and macroeconomic predictors ($model_{r,v,\tilde{x},m}$) remain rather stable over lag structures and only jumps up when 11 lags of the predictors are present in the model.[22] However, the performance of the LSTM models with only lagged returns and volatilities ($model_{r,v}$) keeps increasing and it jumps up when 11 lags of the two predictors are present in the model. These results indicate that one can achieve good performance with an LSTM model that accommodates many firm characteristics. However, it is also possible to develop a simpler LSTM (i.e. $model_{r,v}$)model with only information about the returns and volatilities and achieve performance as good as an LSTM model with a full set of firm characteristics and macroeconomic variables. Finally, we find that macroeconomic predictors with any number of lags do not add additional improvement in volatility prediction when other predictors are present in the model.

[21] Some of the firm-specific predictors are in annual frequency. These variables didn't help in predicting volatility in the models in the previous sections. We assume that their lags also cannot have significant predictive power for the next month volatility and thus drop them to speed up the training of the LSTM model.

[22] In $model_{r,v,\tilde{x},m}$, we let r to stand for return, v for volatility, $\tilde{x}$ for firm-specific predictors other than the return and volatility, and m for macroeconomic predictors.

To further investigate the performance of the LSTM model we fix the number of lags to be 11 and evaluate its performance across several other specifications. First, we change the number of hidden layers from 1 to 4. Second, we increase the number of predictors as explained before. Finally, we compare the results with the FF neural network models. The FF models come with the same specifications except that it contains only 1 lag of predictors. Table 3 shows the out-of-sample results with the MARE and R^2 measures.

[Insert Table 3 and Figure 12 here]

We note that the performance doesn't necessarily improve across all specifications when we increase the number of hidden layers (in the columns) from 1 to 4. However, the performance improves for both FF and LSTM when more predictors are added to the models (in the rows), with the exception being the macroeconomic predictors. Most importantly, we note that the LSTM model outperforms the FF model across all specifications. Figure 12 also provides a visual presentation of these findings but adds also the LSTM models with 1 and 3 lags. The left column is dedicated to the MARE and the right column presents the R^2.

So far all our results in this section have been based on the aggregate performance across subsamples. In Figure 13, we unroll the performance measures over time for the selected competing models from Table 3. More precisely, we compare the dynamics of performance measures for the LSTM $model_{r,v}$ and the LSTM $model_{r,v,\tilde{x}}$ both with 4 hidden layers and 11 lags. We observe that the two LSTM models closely follow each other except that the R^2 for the $model_{r,v}$ drops during the dot-com bubble and financial crisis in 2008-09. This indicates that during the market meltdowns a function of only two predictors might not best predict volatility. Put it differently, only two predictors do not afford to explain changes in the volatility when market complexities are at the highest.

[Insert Figure 13 here]

4 Conclusion

In this paper, we use machine learning methods such as the elastic net model, gradient boosting regression tree model and two variants of neural networks models namely the feedforward and LSTM neural networks for predicting the volatility of stock returns.

We find that a limited number of predictors, namely, the current realized volatility, idiosyncratic volatility, bid-ask spread and return, respectively, account for most of the predictive power of the models. Indeed, these are the same variables that are empirically found in other studies to be most important for predicting expected returns, confirming the risk-returns trade-off in finance. We also find that machine learning methods not only properly capture the order and magnitude of the impact for each predictor but also the direction of the impact as well as the interactions between predictors without any pre-specified interaction effects in the models. In particular, our results indicate that machine learning methods can account for the stylized facts about volatility. We find that large values of the current period volatility have large and positive impact on the next period volatility, whereas, small values of the current period volatility have small and negative impact on the next period volatility. Furthermore, we find that returns have an asymmetric impact on the next period volatility. Negative returns have a higher impact on the volatility than positive returns of the same size. As for the interactions, we note that large (small) bid-ask spreads interact with large (small) values of volatility and have a large (small) and positive (negative) impact on the next period volatility.

Finally, we show that the LSTM model, which captures the temporal dependence of predictors, can outperform the feedforward neural network and regression tree, which rely on the most recent information in the predictors. In particular, our LSTM model with only volatility and return as predictors up to 1 year into the past, performs as good as an LSTM model with the full set of predictors and the same number of lags. One can think of our LSTM model as an alternative to the GARCH type models except that we do not need to impose any distributional assumption.

References

Andersen, T., Bollerslev, T., Christoffersen, P., Diebold, F., 2006. Volatility and correlation forecasting. in G. Elliot, C.W.J. Granger, and A. Timmermann (eds.), Handbook of Economic Forecasting , 778–878.

Andersen, T., Bollerslev, T., Diebold, F., 2003a. Parametric and nonparametric measurements of volatility. in Aït-Sahalia, Y., Hansen, L.P. (eds.), Handbook of Financial Econometrics .

Andersen, T., Bollerslev, T., Diebold, F., Labys, P., 2003b. Modeling and forecasting realized volatility. Econometrica 71, 579–625.

Azinovic, M., Gaegauf, L., Scheidegger, S., 2019. Deep equilibrium nets. Working paper .

Bashchenko, O., Marchal, A., 2019. Deep learning, jumps, and volatility bursts. Swiss Finance Institute Research Paper 20-10.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics 31, 307–327.

Breen, W., Glosten, L., Jagannathan, R., 1989. Economic significance of predictable variations in stock index returns. Journal of Finance 44, 1177–1189.

Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A.W., Siddique, A., 2016. Risk and risk management in the credit card industry. Journal of Banking and Finance 72, 218–239.

Campbell, J., 1987. Stock returns and the term structure. Journal of Financial Economics 18, 373–399.

Campbell, J., Lo, A., MacKinlay, A., 1997. The econometrics of financial markets. Princeton University Press, Princeton .

Caruana, R., Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. ICML 06, 161–168.

Chen, L., Pelger, M., Zhu, J., 2019. Deep learning in asset pricing. Working paper .

Christiansen, C., Schmeling, M., Schrimpf, A., 2012. A comprehensive look at financial volatility prediction by economic variables. Journal of Applied Econometrics 27, 956–977.

David, A., Veronesi, P., 2013. What ties return volatilities to price valuations and fundamentals. Journal of Political Economy 121, 682–746.

Duarte, V., 2017. Machine learning for continuous time finance. Working paper .

Engle, R., 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. Econometrica 50, 987–1008.

Engle, R., Ghysels, E., Sohn, B., 2007. On the economic sources of stock market volatility. Working Paper .

Engle, R., Patton, A., 2001. What good is a volatility model? Quantitative Finance 1, 237–245.

Engle, R., Rangel, R., 2008. The Spline-GARCH model for low-frequency volatility and its global macroeconomic causes. Review of Financial Studies 21, 1187–1222.

French, K., Schwert, G., Stambaugh, R., 1987. Expected stock returns and volatility. Journal of Financial Economics 19, 3–29.

Freyberger, J., Neuhierl, A., Weber, M., 2020. Dissecting characteristics nonparametrically. Review of Financial Studies 33, 2326–2377.

Friedman, J., 2001. Greedy function approximation: A gradient boosting machine. Annals of Statistics 5, 1189–1232.

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: A statistical view of boosting. Annals of Statistics 28, 337–374.

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., Walther, A., 2017. Predictably unequal? The effects of machine learning on credit markets. Working paper .

Glosten, L.R., Jagannathan, R., Runkle, D.E., 1993. On the relation between the expected value and the volatility of the nominal excess returns on stocks. Journal of Finance 48, 1779–1801.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. Review of Financial Studies 33, 2223–2273.

Heaton, J.B., Polson, N.G., Witte, J.H., 2016. Deep learning in finance. Preprint .

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

Hutchinson, J.M., Lo, A.W., Poggio, T., 1994. A nonparametric approach to pricing and hedging derivative securities via learning networks. Journal of Finance 49, 851–859.

Kelly, B.T., Pruitt, S., Yinan, S., 2019. Characteristics are covariances: A unified model of risk and return. Journal of Financial Economics 134, 501–524.

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

Kolm, P.N., Ritter, G., 2019. Dynamic replication and hedging: A reinforcement learning approach. The Journal of Financial Data Science 1, 159–171.

Kozak, S., Nagel, S., Santosh, S., 2020. Shrinking the cross section. Journal of Financial Economics 135, 271–292.

Ludvigsona, S.C., Ng, S., 2007. The empirical risk–return relation: A factor analysis approach. Journal of Financial Economics 83, 171–222.

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I., 2020. From local explanations to global understanding with explainable ai for trees. Nature Machine Intelligence 2, 2522–5839.

Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Mandelbrot, B.B., 1963. The variation of certain speculative prices. The Journal of Business 36, 394–419.

Marquering, W., Verbeek, M., 2004. The economic value of predicting stock index returns and volatility. Journal of Financial and Quantitative Analysis 39, 407–429.

Paye, B.S., 2008. Do macroeconomic variables predict aggregate stock market volatility? Unpublished paper, Rice University .

Poon, S.H., Granger, C.W.J., 2003. Forecasting volatility in financial markets: A review. Journal of Economic Literature XLI, 478–539.

Rossi, A., 2018. Predicting stock market returns with machine learning. Working paper .

Schwert, G., 1989. Why does stock market volatility change over time? Journal of Finance 44, 1115–1153.

Schwert, G.W., Seguin, P.J., 1990. Heteroskedasticity in stock returns. Journal of Finance 45, 1129–1155.

Sirignano, J., Sadhwani, A., Giesecke, K., 2016. Deep learning for mortgage risk. Working Paper, University of Illinois, Urbana-Champaign .

Welch, I., Goyal, A., 2008. A comprehensive look at the empirical performance of equity premium prediction. Review of Financial Studies 21, 1455–1508.

Whitelaw, R., 1994. Time variations and covariations in the expectation and volatility of stock market returns. The Journal of Finance 49, 515–541.

Yao, J., Li, Y., Tan, C.L., 2000. Option price forecasting using neural networks. Omega 28, 455–466.

Table 1: Number of Parameters in the FF and LSTM Neural Networks

		FF			LSTM		
		2 input predictors	46 input predictors	54 input predictors	2 input predictors	46 input predictors	54 input predictors
		I	II	III	IV	V	VI
1-layer	2 units	6	94	110	40	392	456
2-layers	4 units 2 units	22	198	230	168	872	1000
3-layers	8 units 4 units 2 units	70	422	486	616	2024	2280
4-layers	16 units 8 units 4 units 2 units	230	934	1062	2280	5096	5608

Note: This table presents the number of parameters in the feedforward (FF) and LSTM neural networks when the number of units, layers and predictors increases. The output layer is excluded from the calculation.

Table 2: Out-of-sample Performance of the Models

	ELN	GBRT	FF_1	FF_2	FF_3	FF_4
R^2 (%)	61.0	77.4	77.0	77.0	77.3	77.4
RMSE	0.0083	0.0065	0.0058	0.0057	0.0056	0.0056
MSRE	0.1042	0.0494	0.0481	0.0456	0.0452	0.0443
MARE	0.2367	0.1662	0.1637	0.1588	0.1599	0.1568

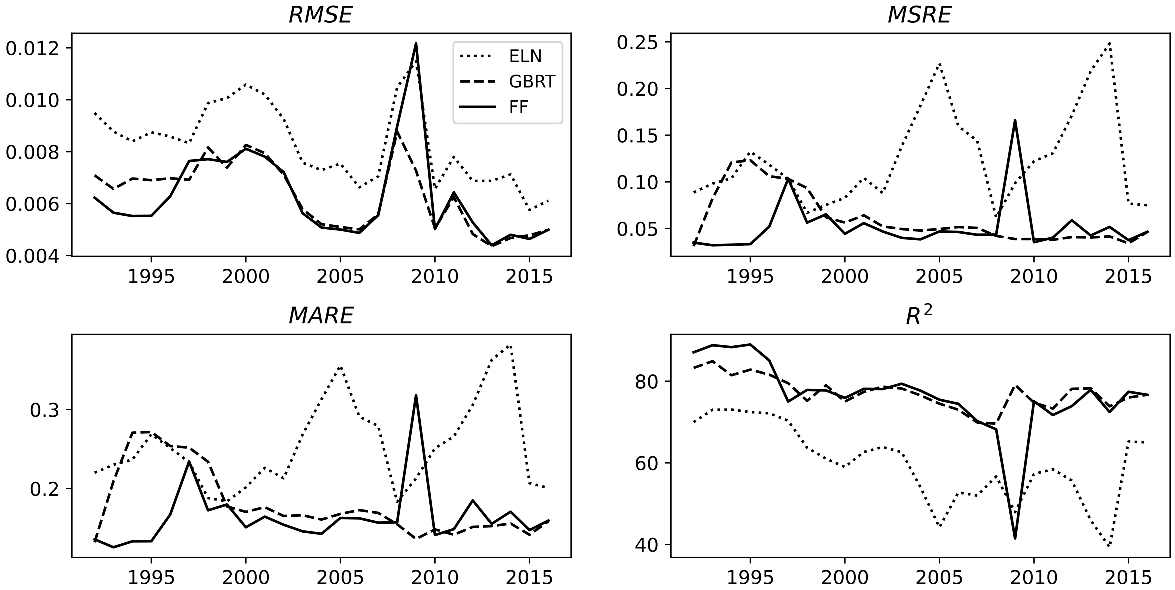
Note: This table presents out-of-sample root mean squared error (RMSE), mean squared relative error (MSRE), mean absolute relative error (MARE) and R^2 (in %) for the elastic net (ELN), gradient boosting regression tree (GBRT), and finally feedforward (FF) neural network model with one to four hidden layers ($FF_l, l = 1, ..., 4$) with the full set of predictors. The numbers are medians across all subsamples. For details about the sample split scheme see Section 3.1.

Table 3: Out-of-sample MARE and R^2 for the FF and LSTM Neural Networks

		1-layer		*2-layers*		*3-layers*		*4-layers*	
		FF	LSTM	FF	LSTM	FF	LSTM	FF	LSTM
$model_{r,v}$	R^2	68.4	76.4	68.6	81.0	68.7	80.3	69.0	**81.8**
	MARE	0.182	0.139	0.178	**0.121**	0.179	0.125	0.179	0.122
$model_{r,v,\tilde{x}}$	R^2	76.1	82.2	76.6	82.9	77.0	**83.1**	77.0	83.1
	MARE	0.164	0.114	0.157	**0.112**	0.156	0.114	0.157	0.115
$model_{r,v,\tilde{x},m}$	R^2	77.0	76.8	77.0	77.2	77.3	**80.2**	77.4	80.0
	MARE	0.164	0.142	0.159	0.144	0.160	**0.128**	0.157	0.136

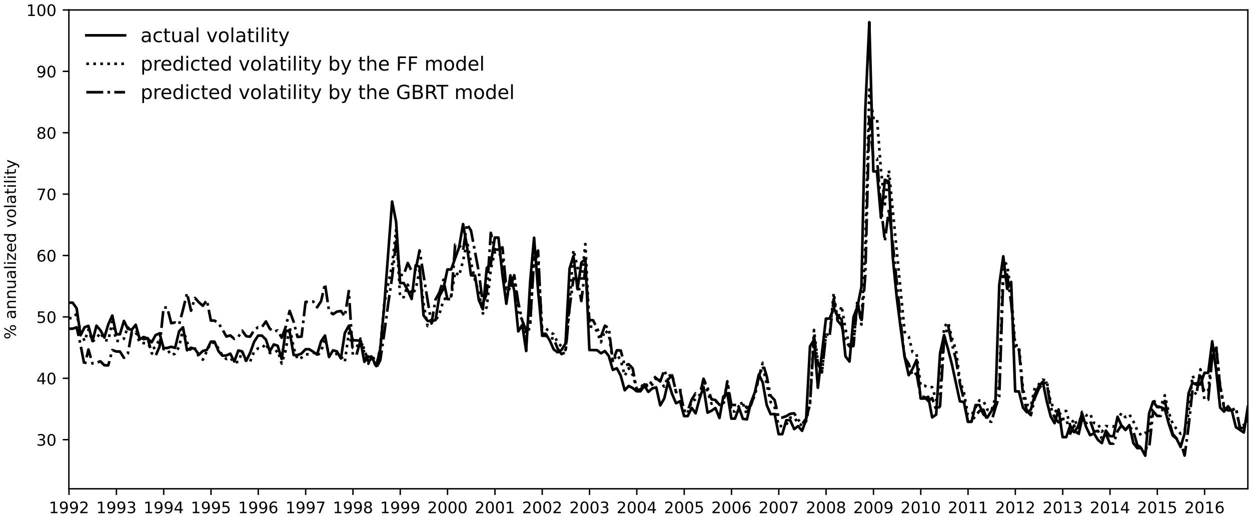
Note: This table presents the evolution of the out-of-sample mean absolute relative error (MARE) and R^2 (in %) for the feedforward (FF) and LSTM neural networks across the 1 to 4 hidden layers in each model. The *models* in each row have different number of predictors. For instance, $model_{r,v,\tilde{x},m}$ contains past returns (r), past volatilities (v), past firm characteristics ($\tilde{x}$) and finally past macroeconomic variables (m) as predictors. Results are shown for the FF and LSTM models with 1 and 11 lags, respectively. The numbers are medians across all subsamples. For details about the sample split scheme see Section 3.1.

Figure 1: Evolution of the Out-of-sample Performance of the Models Over Time



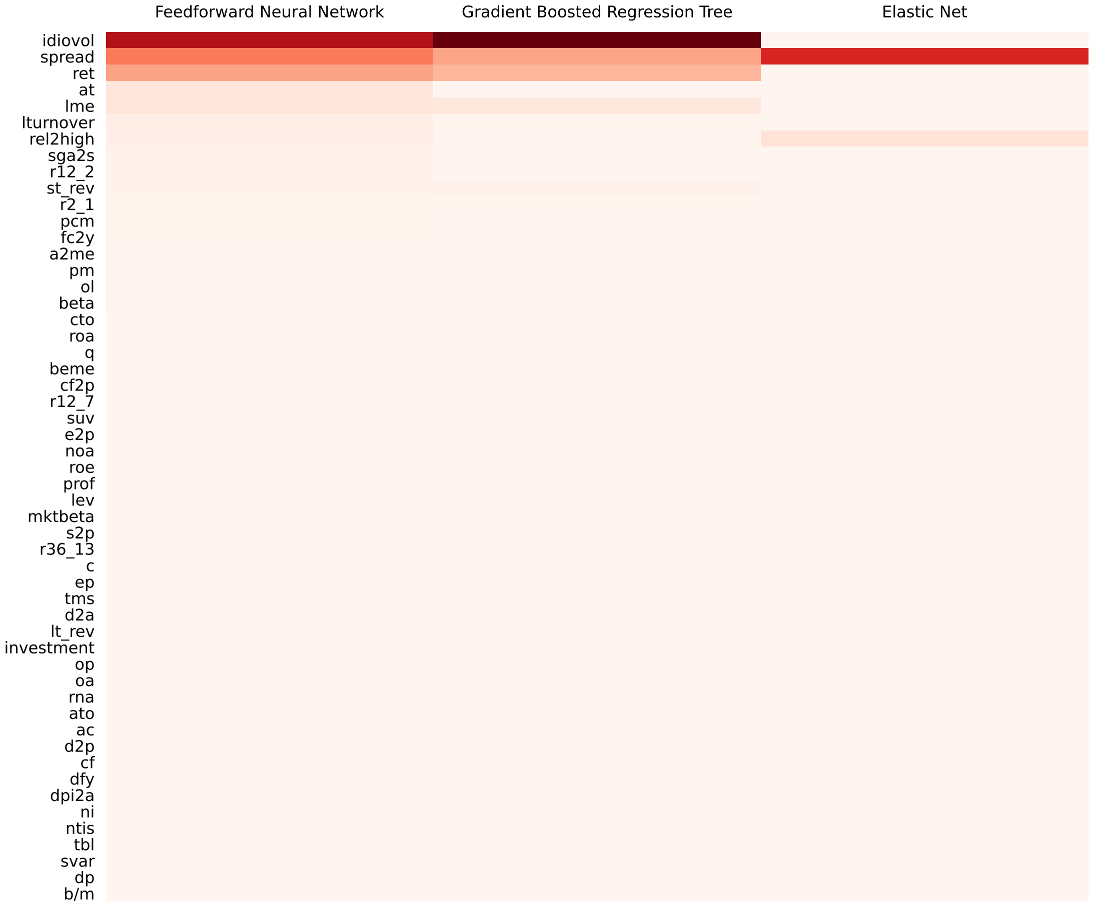
Note: This figure displays the evolution of the out-of-sample root mean squared error (RMSE), mean squared relative error (MSRE), mean absolute relative error (MARE) and R^2 (in %) for the elastic net (ELN), gradient boosting regression tree (GBRT) and feedforward (FF) neural network model with the full set of predictors over the test samples from 1992 to 2016. For details about the sample split scheme see Section 3.1.

Figure 2: Actual versus Out-of-sample Predictions of Realised Volatility



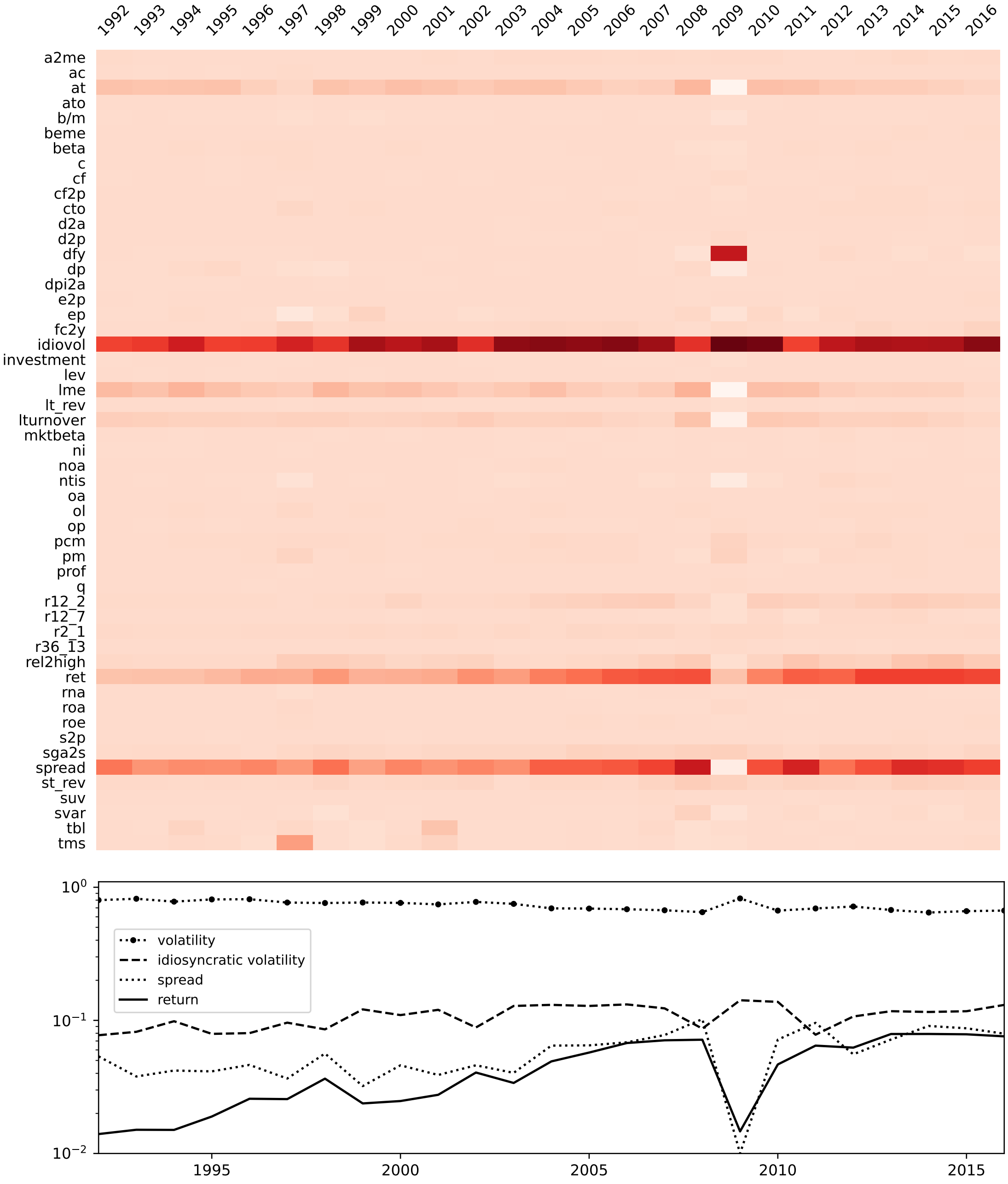
Note: This figure displays the evolution of the monthly out-of-sample predictions of realised volatility for a large cross-section of US stocks using the FF model (dotted line) with four hidden layers, and the GBRT model (dashed line) versus the actual realized volatility (solid line) over the test samples from 1992 to 2016. For each month we take the average of the predictions of (and actual) volatility for the firms available in that month. Predictions by both models are based on all predictors. For details about the sample split scheme see Section 3.1.

Figure 3: Out-of-sample Importance of Predictors for the Three Models



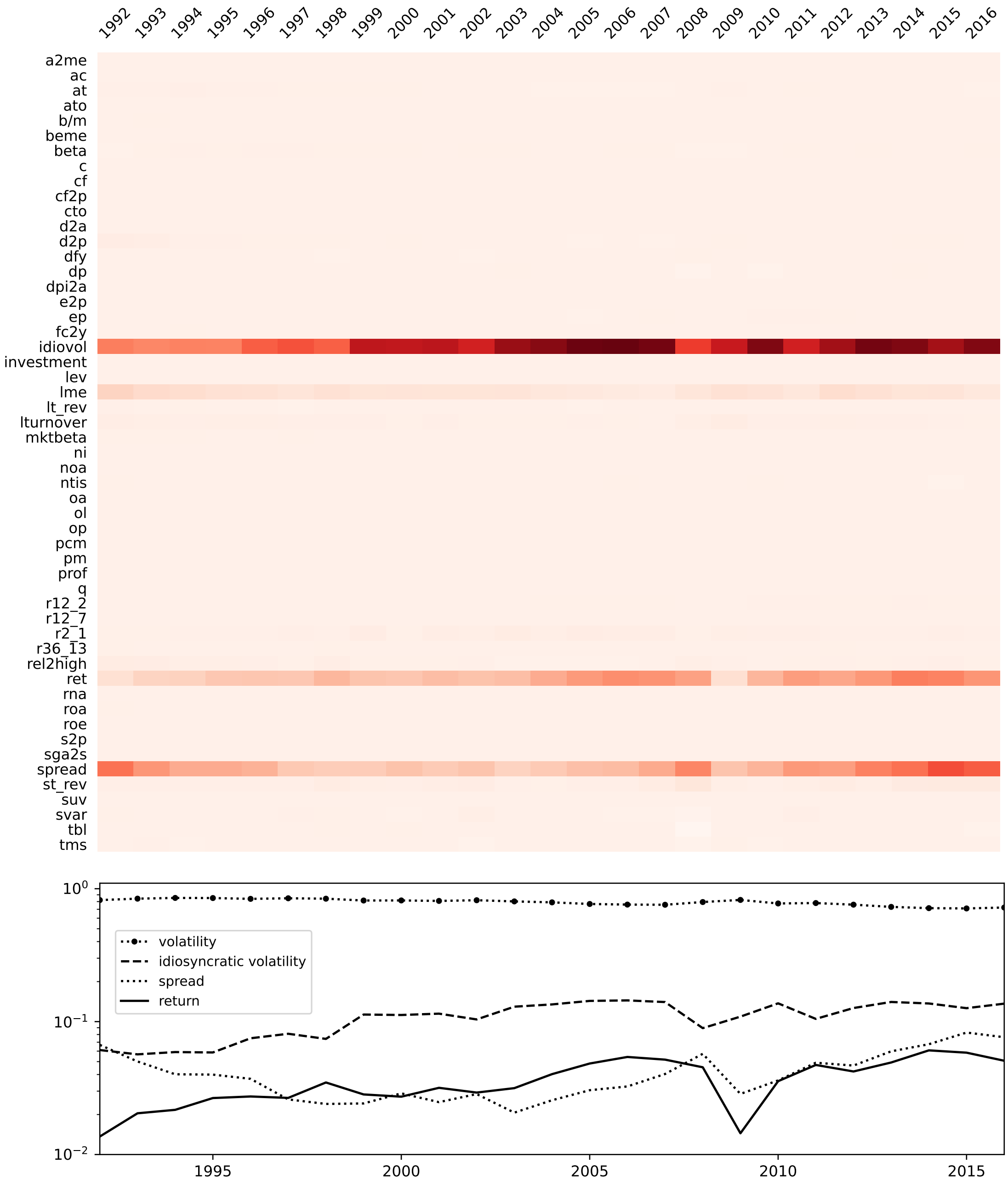
Note: This figure displays the out-of-sample importance scores for the feedforward (FF) neural network, gradient boosted regression tree (GBRT) and the elastic net (ELN) models. The vertical axis shows the importance scores of the 54 predictors sorted for the FF neural network model. Volatility as a predictor is excluded from the list to provide better visualisation of other effects. Importance scores are median across subsamples. For details about the sample split scheme see Section 3.1. Details on the names and definitions of predictors can be found in Appendix A.

Figure 4: Evolution of the Out-of-sample Predictors' Importance for the FF Neural Network Model



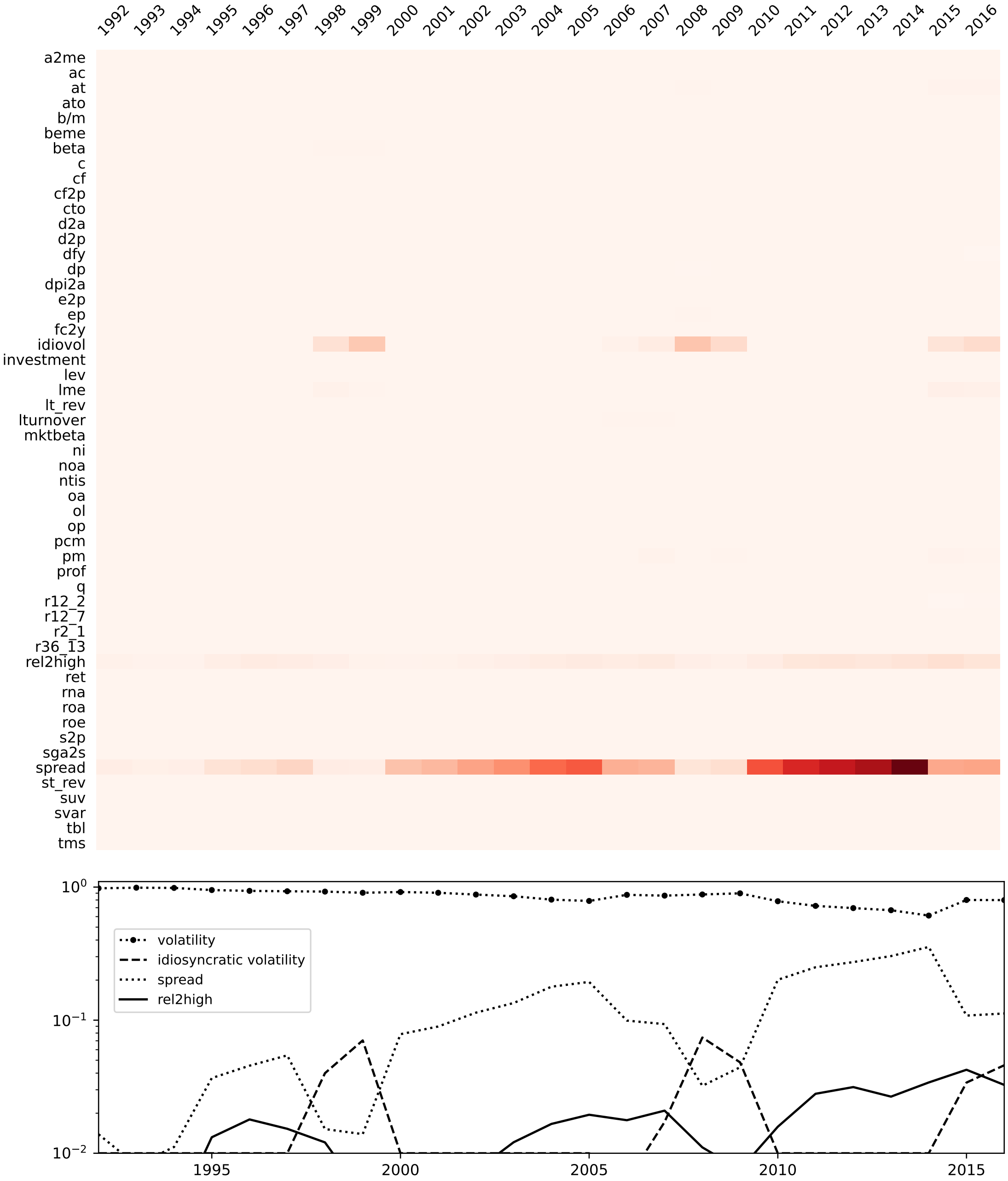
Note: This figure displays the evolution of the out-of-sample importance scores for the feedforward neural network model. The plot on the top shows the importance scores for all predictors. However, volatility as a predictor is excluded from this plot to provide better visualisation of other effects. The plot on the bottom displays the importance scores in the log scale for the top four predictors. For details about the sample split scheme see Section 3.1. Details on the names and definitions of predictors can be found in Appendix A.

Figure 5: Evolution of the Out-of-sample Predictors' Importance for the GBRT Model



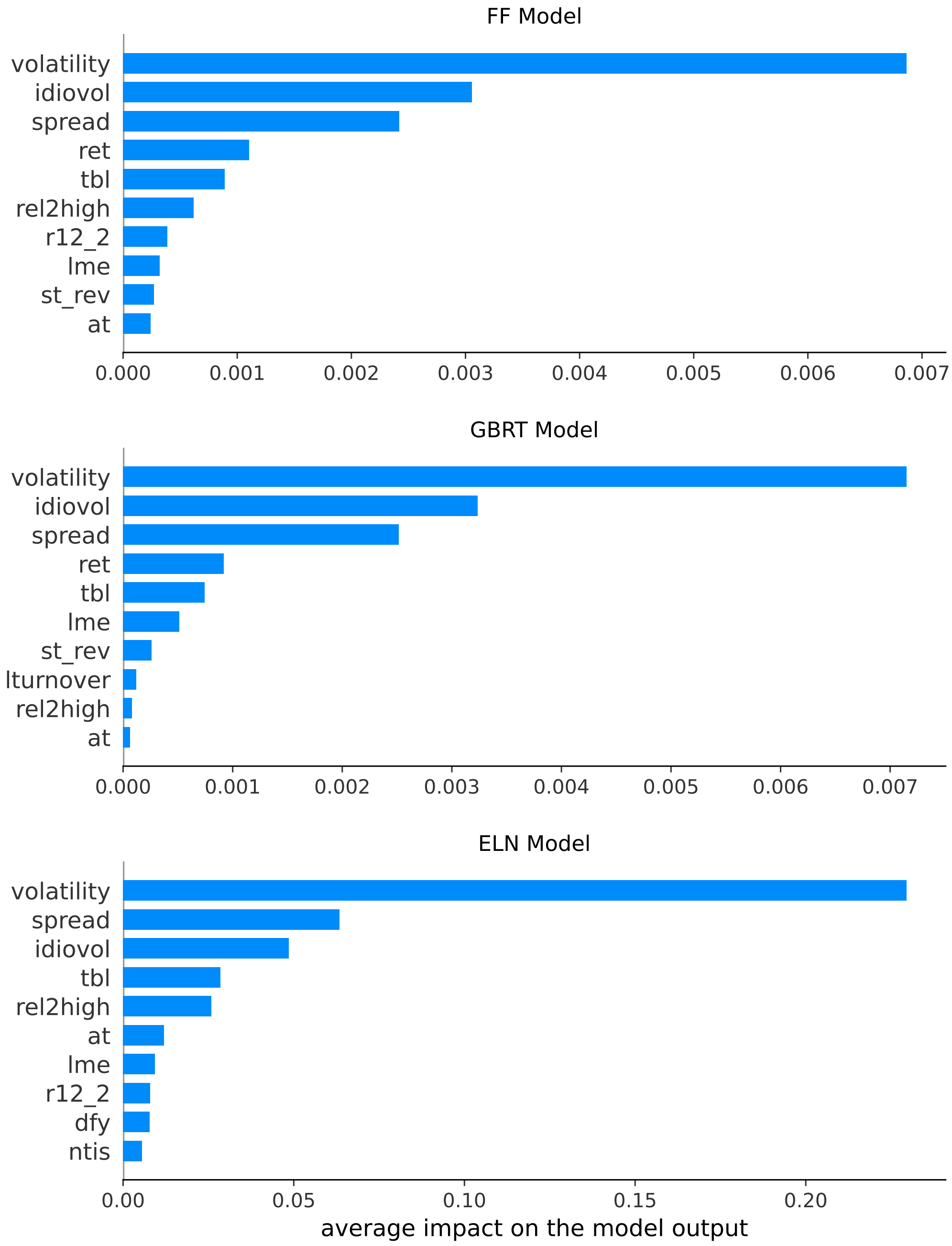
Note: This figure displays the evolution of the out-of-sample importance scores for the gradient boosted regression tree model. The plot on the top shows the importance scores for all predictors. However, volatility as a predictor is excluded from this plot to provide better visualisation of other effects. The plot on the bottom displays the importance scores in the log scale for the top four predictors. For details about the sample split scheme see Section 3.1. Details on the names and definitions of predictors can be found in Appendix A.

Figure 6: Evolution of the Out-of-sample Predictors' Importance for the ELN Model



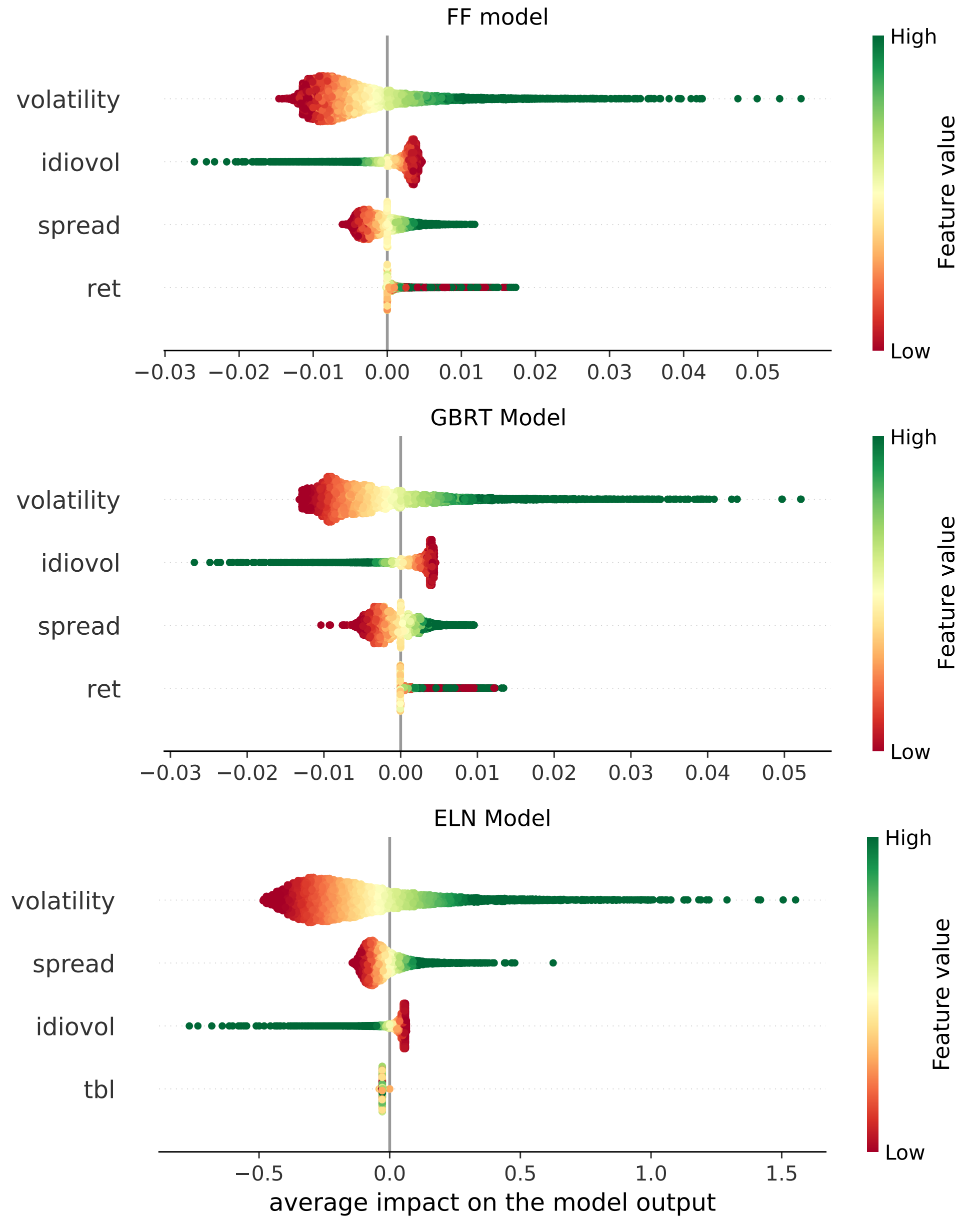
Note: This figure displays the evolution of the out-of-sample importance scores for the elastic net model. The plot on the top shows the importance scores for all predictors. However, volatility as a predictor is excluded from this plot to provide better visualisation of other effects. The plot on the bottom displays the importance scores in the log scale for the top four predictors. For details about the sample split scheme see Section 3.1. Details on the names and definitions of predictors can be found in Appendix A.

Figure 7: Order and Magnitude of the Impact by Top 10 Predictors



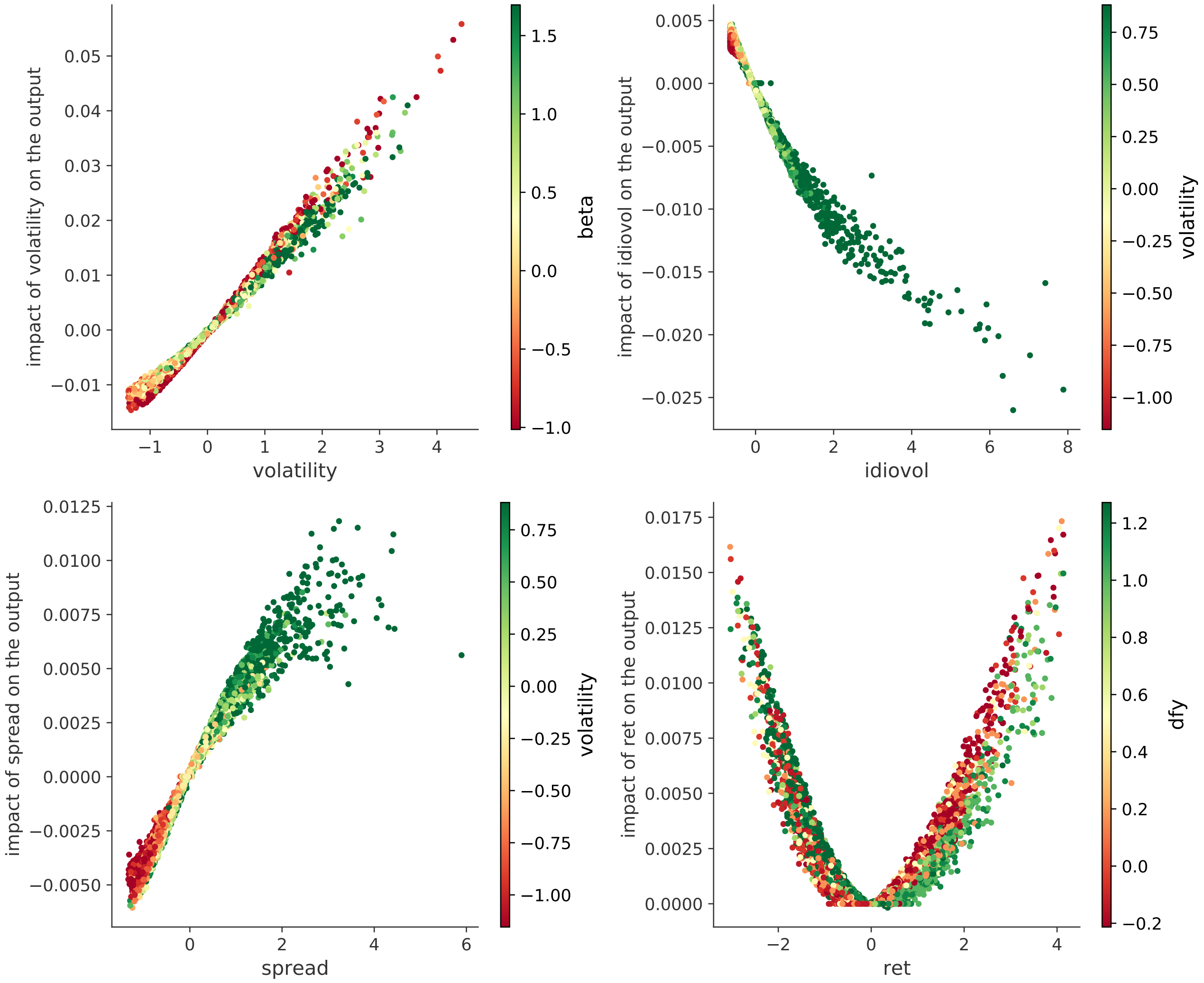
Note: This figure displays the out-of-sample order (on the vertical axis) and magnitude (on the horizontal axis) of the impact of the top 10 predictors on the next period volatility for the three models. Details on the names and definitions of predictors can be found in Appendix A.

Figure 8: Direction and Magnitude of the Impact by Top 4 Predictors



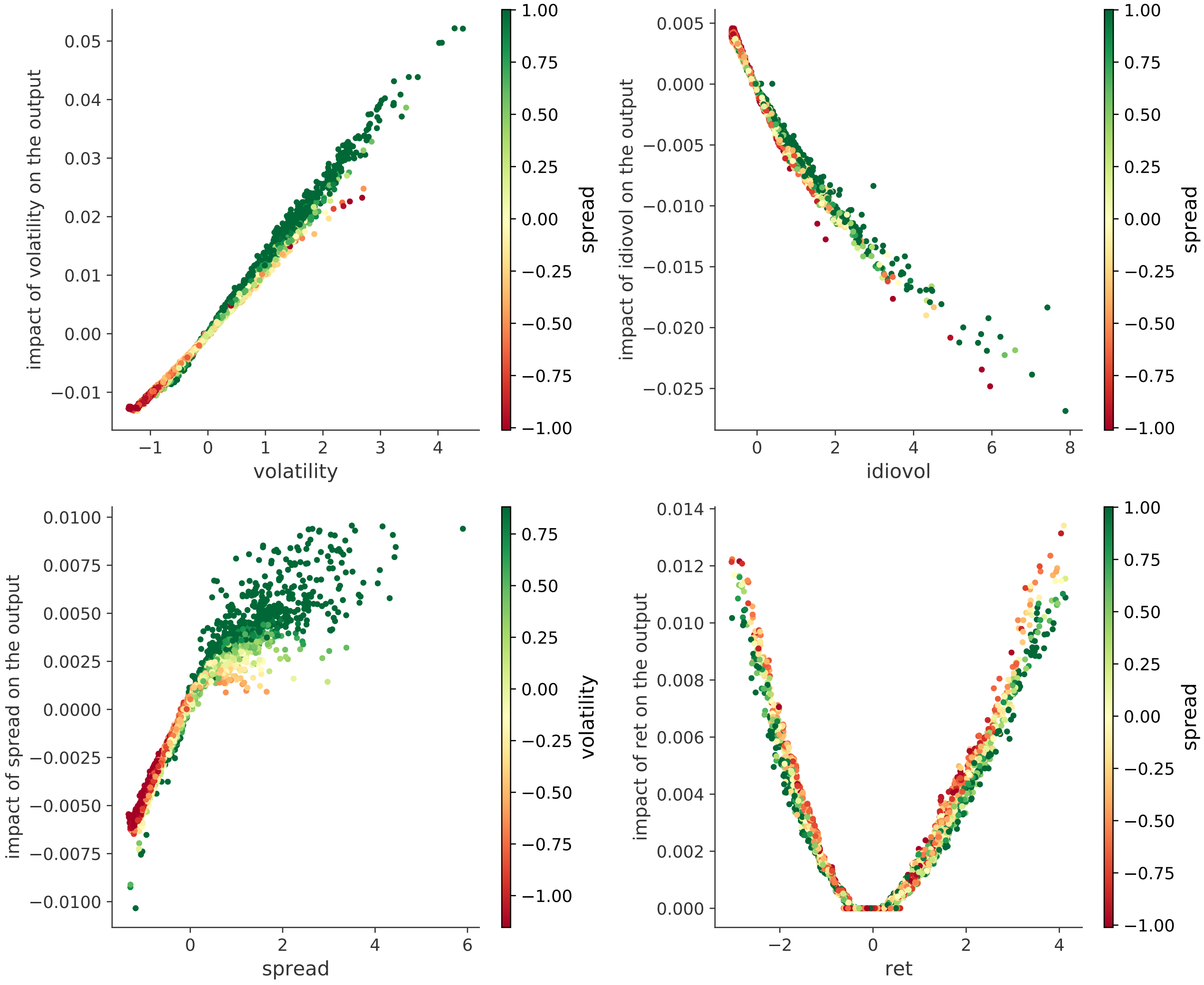
Note: This figure displays the out-of-sample impact of the top 4 predictors on the next period volatility for the three models. The horizontal axis shows the magnitude and direction of impact. The colours from red to green show the magnitude of the predictors' values from low to high. The vertical thickness of each line is due to the overlapping values of the impact. Details on the names and definitions of predictors can be found in Appendix A.

Figure 9: Interaction Effects for the Top 4 Predictors in the FF Neural Network Model



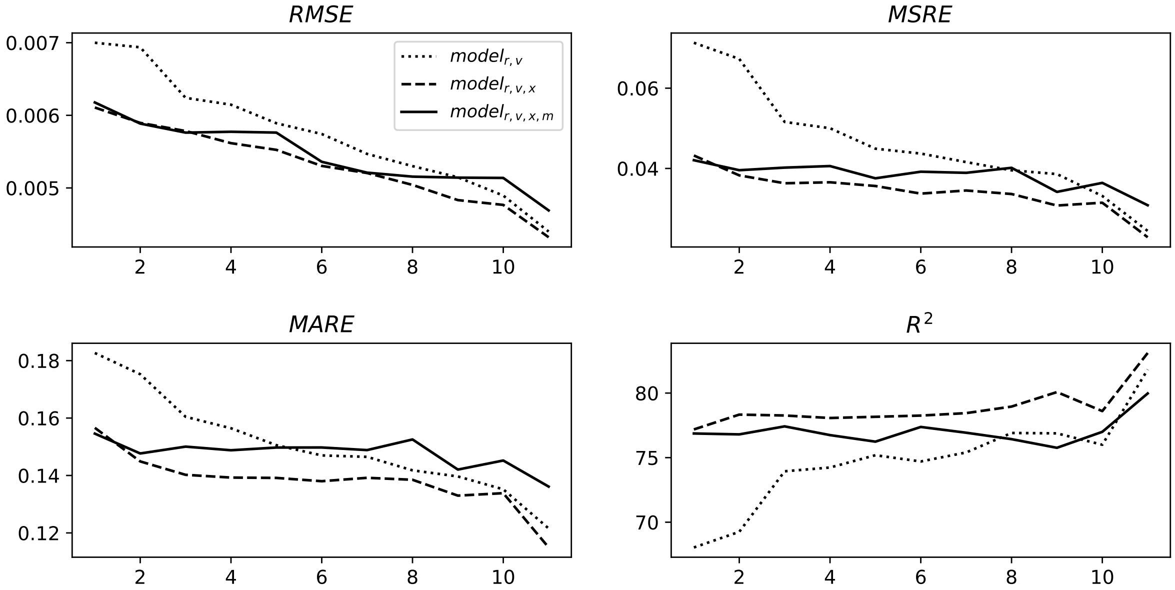
Note: This figure displays the out-of-sample impact of the top 4 predictors (on the horizontal axis) on the next period volatility for the feedforward neural network model. The vertical axis on the right shows (via colours) a predictor which has most interacted with the selected predictor on the horizontal axis, whereas the vertical axis on the left shows impact on the next period volatility. Details on the names and definitions of predictors can be found in Appendix A.

Figure 10: Interaction Effects for the Top 4 Predictors in the GBRT Model



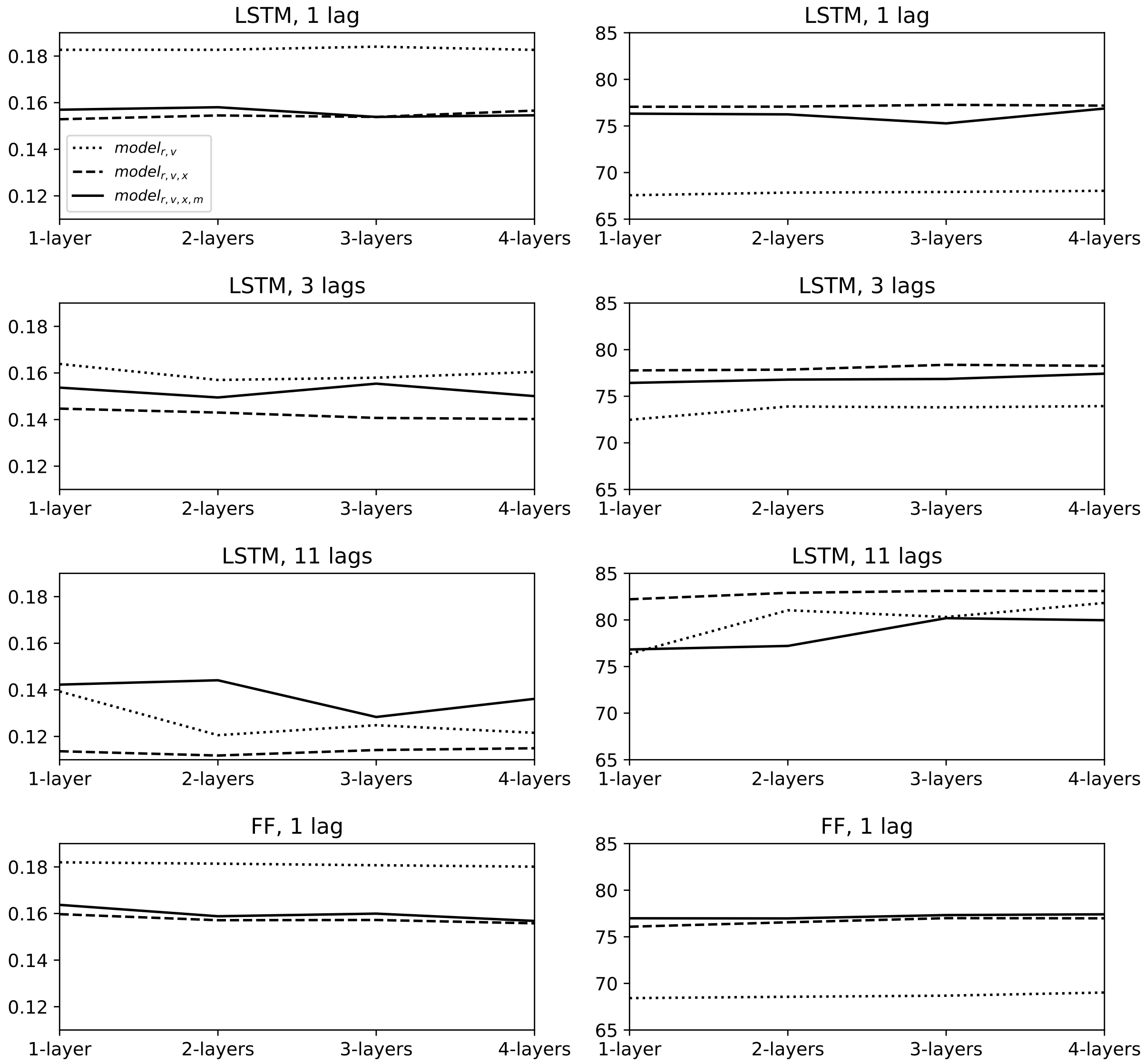
Note: This figure displays the out-of-sample impact of the top 4 predictors (on the horizontal axis) on the next period volatility for the gradient boosted regression tree model. The vertical axis on the right shows (via colours) a predictor which has most interacted with the selected predictor on the horizontal axis, whereas the vertical axis on the left shows impact on the next period volatility. Details on the names and definitions of predictors can be found in Appendix A.

Figure 11: Evolution of the Out-of-sample Performance of the LSTM Models Across Lags



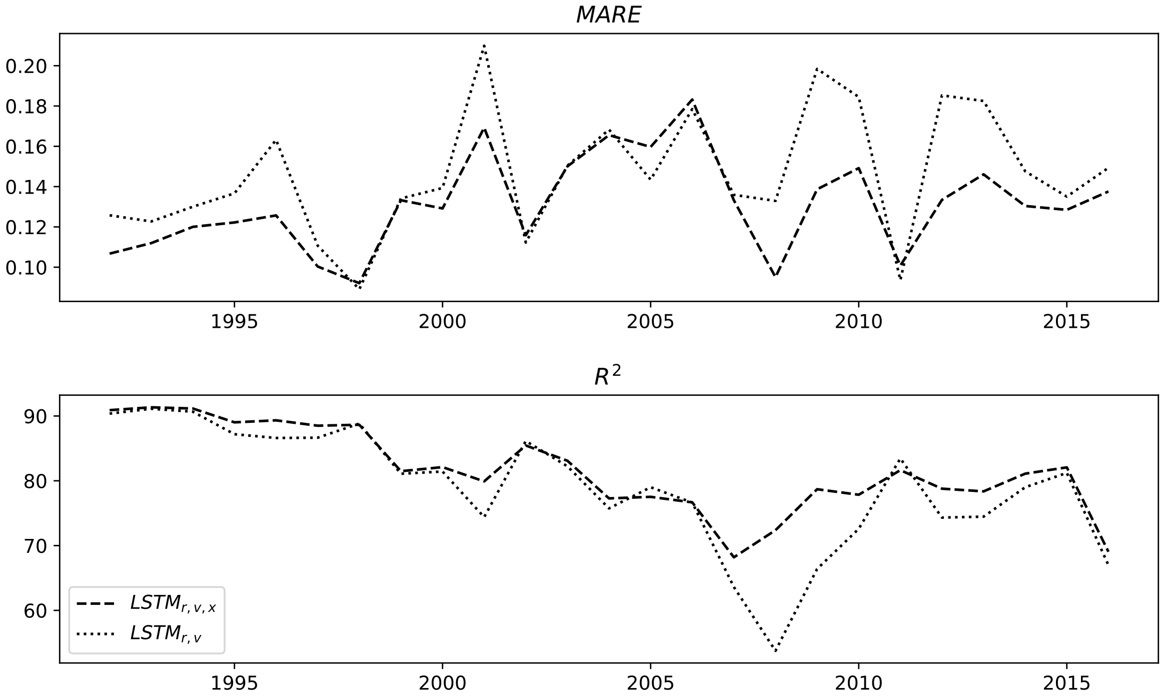
Note: This figure displays the evolution of the out-of-sample root mean squared error (RMSE), mean squared relative error (MSRE), mean absolute relative error (MARE) and R^2 (in %) across 11 lags for the LSTM neural networks model with 4 hidden layers. The dotted line, dashed line and the solid lines represent the models with different number of predictors. For instance, the $model_{r,v,\tilde{x},m}$ contains past returns (r), past volatilities (v), past firm characteristics (x) and finally past macroeconomic variables (m) as predictors.

Figure 12: Evolution of the Out-of-sample Performance of the FF and LSTM Models Across Layers



Note: This figure displays the evolution of the out-of-sample mean absolute relative error (MARE) in the left column and R^2 in percentage in the right column for the feedforward (FF) and LSTM neural networks models across the number of hidden layers in each model. The dotted line, dashed line and solid lines represent the models with a different number of predictors. For instance, the $model_{r,v,\tilde{x},m}$ contains past returns (r), past volatilities (v), past firm characteristics $(\tilde{x})$ and finally past macroeconomic variables (m) as predictors. Results are shown for the selected LSTM models with 1, 3 and 11 lags.

Figure 13: Evolution of the Out-of-sample Performance of the LSTM Models Over Time



Note: This figure displays the evolution of the out-of-sample mean absolute relative error (MARE) and R^2 (in %) for the selected LSTM neural networks models over the test samples from 1992 to 2016. The LSTM models are the selected models based on their performance measures from Table 3. Both models contain 11 lags but a different number of predictors. For details about the sample split see Section 3.1.

Appendices

A Predictors

We describe the firm characteristics and macroeconomic variables that are used as predictors. For an overview see Table A.1.

A.1 Macroeconomic Variables

Dividend Price Ratio (DP): it is the difference between the log of dividends and the log of prices. Dividends are 12-month moving sums of dividends paid on the S&P 500 index.

Earnings Price Ratio (EP): it is the difference between the log of earnings and the log of prices. Earnings are 12-month moving sums of earnings on the S&P500 index.

Book-to-Market Ratio (BM): it is the ratio of book value to market value for the Dow Jones Industrial Average.

Net Equity Expansion (NTIS): it is the ratio of 12-month moving sums of net issues by NYSE listed stocks divided by the total end-of-year market capitalization of NYSE stocks.

Treasury bills (TBL): the yields on short term United States securities.

Term Spread (TMS): it is the difference between the long term yield on government bonds and the Treasury-bill.

Default Yield Spread (DFY): it is the difference between BAA and AAA-rated corporate bond yields.

Stock Variance (SVAR): it is computed as sum of squared daily returns on the S&P 500.

Table A.1: List of Firm Characteristics and Macroeconomic Variables

	Past Returns				**Value**		
(1)	r_{2-1}	(m)	short term momentum	(27)	A2ME	(y)	total assets to Size
(2)	r_{12-2}	(m)	momentum	(28)	BEME	(y)	book to market ratio
(3)	r_{12-7}	(m)	intermediate momentum	(29)	CF	(y)	free cash flow to BE
(4)	r_{36-13}	(m)	long term momentum	(30)	C	(y)	cash to AT
(5)	ST-Rev	(m)	short term reversal	(31)	E2P	(y)	income before extraordinary items to Size
(6)	LT-Rev	(m)	long term reversal	(32)	Q	(y)	tobin's Q
(7)	ret	(m)	stock return	(33)	S2P	(y)	sales to price
				(34)	CF2P	(y)	cash flow to price
	Investment			(35)	D2P	(y)	dividend yield
(8)	Investment	(y)	% change in AT	(36)	Lev	(y)	leverage
(9)	NI	(y)	% change in shares outstanding				
(10)	ΔPI2A	(y)	change in PP&E and inventory over lagged AT		**Trading Frictions**		
(11)	NOA	(y)	non-operating assets over lagged AT	(37)	AT	(y)	total assets
				(38)	Beta	(m)	CAPM beta
	Profitability			(39)	MktBeta	(m)	market beta
(12)	ATO	(y)	sales to lagged net operating assets	(40)	LTurnover	(m)	turnover
(13)	CTO	(y)	sales to lagged total assets	(41)	Idio vol	(m)	idiosyncratic volatility
(14)	OP	(y)	operating profitability	(42)	LME	(m)	size
(15)	FC2Y	(y)	fixed costs to sales	(43)	Rel2high	(m)	closeness to past year high
(16)	SGA2S	(y)	selling, general and administrative expenses	(44)	Spread	(m)	average daily bid-ask spread
(17)	ROE	(y)	income before extraordinary items to lagged BE	(45)	SUV	(m)	standard unexplained volume
(18)	PM	(y)	OI after depreciation over sales	(46)	Volatility	(m)	realized Volatility
(19)	D2A	(y)	capital intensity				
(20)	PROF	(y)	gross profitability over BE		**Macro Variables**		
(21)	RNA	(y)	OI after depreciation to lagged net operating assets	(47)	DP	(m)	divident-price ratio
(22)	ROA	(y)	income before extraordinary items to lagged AT	(48)	EP	(m)	earnings-price ratio
				(49)	BM	(m)	book-to-market ratio
				(50)	NTIS	(m)	net equity expansion
	Intangibles			(51)	TBL	(m)	treasury-bill rate
(23)	AC	(y)	accruals	(52)	TMS	(m)	term spread
(24)	OL	(y)	cost of goods sold+SG&A to total assets	(53)	DFY	(m)	default spread
(25)	PCM	(y)	price to cost margin	(54)	SVAR	(m)	stock variance
(26)	OA	(y)	operating accruals				

Note: 'm' and 'y' represents monthly and yearly, respectively.

A.2 Firm Characteristics

A2ME: We follow Bhandari (1988) and define assets-to-market cap as total assets (AT) over market capitalization as of December $t-1$. Market capitalization is the product of shares outstanding (SHROUT) and price (PRC).

AC: Change in operating working capital per split-adjusted share from the fiscal year end $t-2$ to $t-1$ divided by book equity (defined in BEME) per share in $t-1$. Operating working capital per split-adjusted share is defined as current assets (ACT) minus cash and short-term investments (CHE) minus current liabilities (LCT) minus debt in current liabilities (DLC) minus income taxes payable (TXP).

AT: Total assets (AT) as in Gandhi and Lustig (2015).

ATO: Net sales over lagged net operating assets as in Soliman (2008). Net operating assets are the difference between operating assets and operating liabilities. Operating assets are total assets (AT) minus cash and short-term investments (CHE), minus investment and other advances (IVAO). Operating liabilities are total assets (AT), minus debt in current liabilities (DLC), minus long-term debt (DLTT), minus minority interest (MIB), minus preferred stock (PSTK), minus common equity (CEQ).

BEME: Ratio of book value of equity to market value of equity. Book equity is shareholder equity (SH) plus deferred taxes and investment tax credit (TXDITC), minus preferred stock (PS). SH is shareholders' equity (SEQ). If missing, SH is the sum of common equity (CEQ) and preferred stock (PS). If missing, SH is the difference between total assets (AT) and total liabilities (LT). Depending on availability, we use the redemption (item PSTKRV), liquidating (item PSTKL), or par value (item PSTK) for PS. The market value of equity is as of December t-1. The market value of equity is the product of shares outstanding (SHROUT) and price (PRC). See Rosenberg, Reid, and Lanstein (1985) and Davis, Fama, and French (2000).

Beta: We follow Frazzini and Pedersen (2014) and define the CAPM beta as product of correlations between the excess return of stock i and the market excess return and the ratio of volatilities. We calculate volatilities from the standard deviations of daily log excess returns over a one-year horizon requiring at least 120 observations. We estimate correlations using overlapping three-day log excess returns over a five-year period requiring at least 750 non-missing observations.

C: Ratio of cash and short-term investments (CHE) to total assets (AT) as in Palazzo (2012).

CF: Cash flow to book value of equity is the ratio of net income (NI), depreciation and amortization (DP), less change in working capital (WCAPCH), and capital expenditure

(CAPX) over the book-value of equity defined as in the construction of BEME (see Hou et al. (2011)).

CF2P: Cash flow over market capitalization (PRC*SHROUT) as of December $t-1$. Cash flow is defined as income before extraordinary items (IB) plus depreciation and amortization (DP) plus deferred taxes (TXDB).

CTO: We follow Haugen and Baker (1996) and define capital turnover as ratio of net sales (SALE) to lagged total assets (AT).

D2A: Ratio of depreciation and amortization (DP) to total assets (AT)

D2P: Total dividends (DIVAMT) paid from July of $t-1$ to June of t per dollar of equity (LME) in June of t

ΔPI2A: We define the change in property, plants, and equipment following Lyandres, Sun, and Zhang (2008) as changes in property, plants, and equipment (PPEGT) and inventory (INVT) over lagged total assets (TA).

E2P: We follow Basu (1983) and define earnings to price as the ratio of income before extraordinary items (IB) to the market capitalization as of December $t-1$. Market capitalization is the product of shares outstanding (SHROUT) and price (PRC).

FC2Y: Ratio of selling, general, and administrative expenses (XSGS), research and development expenses (XRD), and advertising expenses (XAD) to net sales (SALE).

Idio vol: Idiosyncratic volatility is the standard deviation of the residuals from a regression of excess returns on the Fama and French (1993) three-factor model as in Ang, Hodrick, Xing, and Zhang (2006). We use one month of daily data and require at least fifteen non-missing observations.

Investment: We define investment as the percentage year-on-year growth rate in total assets (AT) following Cooper, Gulen, and Schill (2008).

Lev: Leverage is the ratio of long-term debt (DLTT) and debt in current liabilities (DLC) to the sum of long-term debt, debt in current liabilities, and stockholders' equity (SEQ) following Lewellen (2015).

LME: Size is the total market capitalization of the previous month defined as price (PRC) times shares outstanding (SHROUT) as in Fama and French (1992).

LT Rev: Cumulative return from 60 months before the return prediction to 13 months before.

LTurnover: Turnover is last month's volume (VOL) over shares outstanding (SHROUT) (Datar, Naik, and Radclife (1998)).

MktBeta: Coefficient of the market excess return from the regression on excess returns in the past 60 months (24 months minimum).

NI: The change in the natural log of split-adjusted shares outstanding (CSHO*AJEX) from the fiscal year-end in $t-2$ to the fiscal year-end in $t-1$.

NOA: Net operating assets are the difference between operating assets minus operating liabilities scaled by lagged total assets as in Hirshleifer, Hou, Teoh, and Zhang (2004). Operating assets are total assets (AT) minus cash and short-term investments (CHE), minus investment and other advances (IVAO). Operating liabilities are total assets (AT), minus debt in current liabilities (DLC), minus long-term debt (DLTT), minus minority interest (MIB), minus preferred stock (PSTK), minus common equity (CEQ).

OA: We follow Sloan (1996) and define operating accruals as changes in non-cash working capital minus depreciation (DP) scaled by lagged total assets (TA). Non-cash working capital is the difference between non-cash current assets and current liabilities (LCT), debt in current liabilities (DLC) and income taxes payable (TXP). Non-cash current assets are current assets (ACT) minus cash and short-term investments (CHE).

OL: Operating leverage is the sum of cost of goods sold (COGS) and selling, general, and administrative expenses (XSGA) over total assets as in Novy-Marx (2011).

OP: Annual revenues (REVT) minus cost of goods sold (COGS), interest expense (TIE), and selling, general, and administrative expenses (XSGA) divided by book equity (defined in BEME).

PCM: The price-to-cost margin is the difference between net sales (SALE) and costs of goods sold (COGS) divided by net sales (SALE) as in Gorodnichenko and Weber (2016) and D'Acunto, Liu, Pflueger, and Weber (2017).

PM: The profit margin is operating income after depreciation (OIADP) over sales (SALE) as in Soliman (2008).

PROF: We follow Ball, Gerakos, Linnainmaa, and Nikolaev (2015) and define profitability as gross profitability (GP) divided by the book value of equity as defined above.

Q: Tobin's Q is total assets (AT), the market value of equity (SHROUT times PRC) minus cash and short-term investments (CEQ), minus deferred taxes (TXDB) scaled by total assets (AT).

ret: Current time return

$\boldsymbol{r_{2-1}}$: We define short-term reversal as lagged one-month return as in Jegadeesh (1990).

$\boldsymbol{r_{12-2}}$: We define momentum as cumulative return from 12 months before the return prediction to two months before as in Fama and French (1996).

$\boldsymbol{r_{12-7}}$: We define intermediate momentum as cumulative return from 12 months before the return prediction to seven months before as in Novy-Marx (2012).

$\boldsymbol{r_{36-13}}$: Long-term reversal is the cumulative return from 36 months before the return prediction to 13 months before as in De Bondt and Thaler (1985).

Rel2High: Closeness to 52-week high is the ratio of stock price (PRC) at the end of the previous calendar month and the previous 52 week high price defined as in George and Hwang (2004).

RNA: The return on net operating assets is the ratio of operating income after depreciation to lagged net operating assets (Soliman (2008)). Net operating assets are the difference between operating assets minus operating liabilities. Operating assets are total assets (AT) minus cash and short-term investments (CHE), minus investment and other advances (IVAO). Operating liabilities are total assets (AT), minus debt in current liabilities (DLC), minus long-term debt (DLTT), minus minority interest (MIB), minus preferred stock (PSTK), minus common equity (CEQ).

ROA: Return-on-assets is income before extraordinary items (IB) to lagged total assets (AT) following Balakrishnan, Bartov, and Faurel (2010).

ROE: Return-on-equity is income before extraordinary items (IB) to lagged book-value of equity as in Haugen and Baker (1996).

S2P: Sales-to-price is the ratio of net sales (SALE) to the market capitalization as of December following Lewellen (2015).

SGA2S: SG&A to sales is the ratio of selling, general and administrative expenses (XSGA) to net sales (SALE).

Spread: The bid-ask spread is the average daily bid-ask spread in the previous months as in Chung and Zhang (2014).

STRev: Prior month return

SUV: Standard unexplained volume is difference between actual volume and predicted volume in the previous month. Predicted volume comes from a regression of daily volume on a constant and the absolute values of positive and negative returns. Unexplained volume is standardized by the standard deviation of the residuals from the regression as in Garnkel (2009).

Volatility: Realised volatility of daily returns in the past two months. This predictor is the same as the response variable but it is lagged by one period.

B Importance Scores

The importance scores that lead to our main results, namely the Shapley values, are based on cooperative game theory. Shapley value, first introduced in 1953 by Lloyd Shapley, defines the payoff that each player should expect from her cooperation in a game. In the context of machine learning, Shapley values provide a natural way to compute the contribution of each predictor when predicting the response variable. More precisely, each predictor can appear in multiple coalitions leading to multiple contributions, and the Shapley value is the average of all such contributions.

More formally, the Shapley value for an instance of the predictor p is defined as follows,[23]

$$\varphi_{x_t^p}(f) = \frac{1}{P} \sum_{s \subseteq \{x_t^1,\dots,x_t^P\} \setminus \{x_t^p\}} \binom{P-1}{|s|}^{-1} \underbrace{\left[f(s \cup \{x_t^p\}) - f(s)\right]}_{\text{marginal contribution}} \tag{A.1}$$

where P is the total number of predictors, $|s|$ is the size of the subset of predictors excluding the p^{th} predictor, and f is the estimated function defined in (2). Note that $\binom{P-1}{|s|}$ is the number of possible combinations where s doesn't contain x_t^p.

The marginal contribution of predictor x_t^p is the increase in f as a result of predictor x_t^p joining the subsets s. The Shapley value for predictor x_t^p is the weighted average of its marginal contributions to all possible subsets. To further clarify the formula in (A.1), let's suppose that we have in total three predictors ($P = 3$), and show an instance of these predictors at time t by ret_t, vol_t and an arbitrary predictor x_t^3. We use the formula in (A.1) to calculate the Shapley value $\varphi_{x_t^3}(f)$ for x_t^3. We have $|s| = \{0, 1, 2\}$, so the weights become,

$$\begin{aligned}
s &= \{\emptyset\} & \Rightarrow \quad & \binom{3-1}{0}^{-1} = 1 \\
s &= \{ret_t\} & \Rightarrow \quad & \binom{3-1}{1}^{-1} = 0.5 \\
s &= \{vol_t\} & \Rightarrow \quad & \binom{3-1}{1}^{-1} = 0.5 \\
s &= \{ret_t, vol_t\} & \Rightarrow \quad & \binom{3-1}{2}^{-1} = 1
\end{aligned} \tag{A.2}$$

[23] Note that x is a P-dimensional vector of predictors $x^1, ..., x^P$, and the p^{th} predictor x^p contains instances $x_{i,t}^p$ for stock i at time t. For simplicity reason, in order to present the instance $x_{i,t}^p$, we drop the stock subscript i and let x_t^p present a single instance of the predictor x^p.

and the marginal contributions become,

$$\begin{aligned}
s &= \{\emptyset\} & \Rightarrow \quad && f(x^3) - f(\{\emptyset\}) &= f(\overline{ret_t}, \overline{vol_t}, x^3) - f(\overline{ret_t}, \overline{vol_t}, \overline{x^3}) \\
s &= \{ret_t\} & \Rightarrow \quad && f(ret_t, x^3) - f(ret_t) &= f(ret_t, \overline{vol_t}, x^3) - f(ret_t, \overline{vol_t}, \overline{x^3}) \\
s &= \{vol_t\} & \Rightarrow \quad && f(vol_t, x^3) - f(vol_t) &= f(\overline{ret_t}, vol_t, x^3) - f(\overline{ret_t}, vol_t, \overline{x^3}) \\
s &= \{ret_t, vol_t\} & \Rightarrow \quad && f(ret_t, vol_t, x^3) - f(ret_t, vol_t) &= f(ret_t, vol_t, x^3) - f(ret_t, vol_t, \overline{x^3})
\end{aligned} \tag{A.3}$$

where $f(\overline{ret_t}, \overline{vol_t}, x^3)$ is the value of f at x^3 when ret_t and vol_t are at their average values. Note the function f will not be estimated for each subset s, instead each time a predictor is absent from s, it will be set to its mean value. Combining (A.2) and the marginal contributions from (A.3) we have,

$$\begin{aligned}
\varphi_{x_t^3}(f) =& \frac{1}{3}\Big[1 \times [f(\overline{ret_t}, \overline{vol_t}, x^3) - f(\overline{ret_t}, \overline{vol_t}, \overline{x^3})] \\
&+ 0.5 \times [f(ret_t, \overline{vol_t}, x^3) - f(ret_t, \overline{vol_t}, \overline{x^3})] \\
&+ 0.5 \times [f(\overline{ret_t}, vol_t, x^3) - f(\overline{ret_t}, vol_t, \overline{x^3})] \\
&+ 1 \times [f(ret_t, vol_t, x^3) - f(ret_t, vol_t, \overline{x^3})]\Big].
\end{aligned}$$

Swiss Finance Institute

Swiss Finance Institute (SFI) is the national center for fundamental research, doctoral training, knowledge exchange, and continuing education in the fields of banking and finance. SFI's mission is to grow knowledge capital for the Swiss financial marketplace. Created in 2006 as a public–private partnership, SFI is a common initiative of the Swiss finance industry, leading Swiss universities, and the Swiss Confederation.

swiss:finance:institute

c/o University of Geneva, Bd. Du Pont d'Arve 42, CH-1211 Geneva 4
T +41 22 379 84 71, rps@sfi.ch, www.sfi.ch