International Review of Financial Analysis 96 (2024) 103711

Contents lists available at ScienceDirect

International Review of Financial Analysis

journal homepage: www.elsevier.com/locate/irfa

Macro-Driven Stock Market Volatility Prediction: Insights from a New Hybrid Machine Learning Approach

Qing Zeng [a], Xinjie Lu [b,c,*], Jin Xu [b,c], Yu Lin [a]

[a] School of Business, Chengdu University of Technology, Chengdu, China
[b] School of Economics and Management, Southwest Jiaotong University, Chengdu, China
[c] Service Science and Innovation Key Laboratory of Sichuan Province, China

ARTICLE INFO

Keywords:
Machine learning
Stock market volatility
Macroeconomic variables
Hybrid model
Model explanation
LASSO method

ABSTRACT

This study comprehensively investigates stock market volatility based on over one hundred monthly macroeconomic variables, applying machine learning models. Methodological contribution integrating the random forest (RF) with the least absolute shrinkage and selection operator methods (LASSO). Importantly, the RF-LASSO model can robustly achieve the best forecasting performance under different circumstances. In addition, we focus on model explanation from different perspectives based on permutation importance and shapley additive explanation (SHAP) methods. This study illuminates novel insights into the realm of stock market volatility, harnessing the transformative potential of machine learning methodologies.

1. Introduction

Research on risk management and operations is a hot spot recently because operations research provides enormous tools to deal with risks in a changeable world, including the evolving risk landscape (Aven, 2016; Chen, Kumara, & Sivakumar, 2021; Gala et al., 2023; Pesaran et al., 2009; Sheu, 2007; Wu, 2016; Zwikael & Sadeh, 2006) . Full of uncertainty, dynamics, complexity, and interconnection, risk is everywhere and keeps up with this dynamic environment, which makes it difficult to quantify (Bollerslev et al., 2018; Liu et al., 2023; Power, 2004). Stock market accounts for a large proportion of global wealth, which is essential in the financial market, and modelling its volatility is important for risk management. The issue of stock market volatility is concerned by quantitative researchers and traders (Guiso et al., 2008; Leippold et al., 2022; Liang et al., 2024; Schwert, 1989). Macroeconomic information holds paramount significance in empirical finance, acting as a cornerstone for understanding and predicting financial market dynamics. Macroeconomic fundamentals have been confirmed to have an intimate relationship with stock market (Chortareas & Noikokyris, 2014; Goyal & Welch, 2003; Hung & Yeh, 2023; Razmi et al., 2020). Thus, this study attempts to forecast stock market volatility using macroeconomic information.

In the past few decades, increasing data availability and improving the continuous development of computer technology have triggered the rapid development of algorithms and the field of machine learning (Burkart & Huber, 2021; Rudin & Wagstaff, 2014). Thus, a wide range of machine learning methods are applied in all fields, and the field of stock markets is no exception (Leippold et al., 2022; Rasekhschaffe & Jones, 2019). The existing literature mainly investigates stock market volatility in two ways. The first way is traditional predictive models, such as autoregressive regression (Bollerslev & Mikkelsen, 1996), generalized autoregressive conditional heteroskedasticity (Franses & van Dijk, 1996; Sharma, 2016), the heterogeneous autoregressive model of realized volatility (Liang et al., 2021; Ma, Wei, et al., 2018), generalized autoregressive conditional heteroskedasticity, and mixed data sampling (Fang et al., 2020; Girardin & Joyeux, 2013). The second way is artificial intelligence analysis methods, including support vector machines (Patel et al., 2015), LASSO (Ma et al., 2023), artificial neural networks (Moghaddam et al., 2016), random forest (Ballings et al., 2015; Gupta et al., 2023), decision boosted tree (Christensen et al., 2023; Nayak et al., 2016), XGBoost (Han et al., 2023), deep learning (Ma, Gopalakrishnan, et al., 2018) and some hybrid models (Niu et al., 2024). However, which machine learning model has a relatively better performance for predicting stock market volatility. Thus, this paper tries to make a comparison of the existing commonly used machine learning models from a comprehensive perspective.

Compared with traditional predictive models, machine learning models have some advantages in the forecasting process. First, the

* Corresponding author at: School of Economics and Management, Southwest Jiaotong University, China.
E-mail address: luxinjie@my.swjtu.edu.cn (X. Lu).

https://doi.org/10.1016/j.irfa.2024.103711
Received 7 July 2024; Received in revised form 13 October 2024; Accepted 23 October 2024
Available online 28 October 2024

premise of machine learning algorithms is that they don't need to presuppose any specific functional forms, relational dynamics, or probabilistic distributions (Gu et al., 2020; Watanabe, 2013). Second, they are well-known for handling the nonlinearity in the dataset and capturing valuable information from enormous predictors (Bianchi et al., 2021; Borovkova & Tsiamas, 2019; Chen, Wang, et al., 2021). Thus, in this study, we comprehensively investigate stock market volatility by considering regularization models (LASSO, ridge, elastic net, adaptive LASSO, group LASSO, etc.), tree-based algorithms (random forest, gradient boosting, adaboost, XGBoost, etc.), and neural networks (long short-term memory model, etc.).

Based on the commonly used machine learning models, we empirically find random forest model has a relatively better predictive performance than other commonly used machine learning models, which may be because the random forest model has good anti-noise ability by introducing randomness in the process and applies unbiased estimation for generalization error. However, according to Hastie et al. (2009) and Tang et al. (2018), it has been observed that random forest algorithms can encounter challenges with overfitting when they are applied to datasets characterized by noisy classification or regression tasks. This susceptibility to overfitting is a critical consideration when utilizing random forest model. It is interesting to investigate that whether the forecasting accuracy of random forest model be further improved? It is worth mentioning that LASSO method is good at handling overfitting issues (Lu et al., 2021; Ma et al., 2023; Zhang, Ma, & Wang, 2019). Therefore, we are motivated to combine the random forest and LASSO methods to construct a new method, RF-LASSO, for stock market volatility prediction. Empirical results show that this new hybrid model can robustly achieve a more satisfactory forecasting performance than commonly used machine learning models.

Several findings are obtained. First, most machine learning models can predict stock market volatility. Second, we find that the random forest model has relatively better performance in predicting stock market volatility predictions. Third, the new hybrid model we constructed by combining the random forest and LASSO methods has the best predictive performance, showing that the combination of random forest and LASSO methods can further improve the forecasting accuracy of random forest. Fourth, the robustness checks show that the new hybrid model can perform robustly under different circumstances.

This study's contributions are as follows. First, existing studies have investigated stock market volatility mainly based on some machine learning models. However, as many machine models have been proposed and widely applied, it is interesting to comprehensively investigate which machine learning method is more powerful. Thus, from a comprehensive perspective, this study applies a large set of machine learning methods, including regularization models, tree-based algorithms, and neural networks.

Second, we make methodological innovation by constructing a new hybrid model that integrates the random forest method with the LASSO method. More specifically, we aim not only to perform a complete comparison across machine learning methods but also to focus on how to further increase the accuracy of forecasting volatility based on the existing models. More specifically, the random forest model has relatively better predictive performance than other commonly applied machine learning models for predicting stock market volatility, but it has a weakness in handling overfitting issues. In the meantime, the LASSO method is good at dealing with this issue. Thus, we construct a new hybrid model using a combination of the random forest and LASSO methods. The empirical results show that the new hybrid model further improves the forecasting accuracy of stock market volatility.

Third, we explain how the model system works from different perspectives. For instance, we focus on the model explanation based on permutation importance and shapley additive explanation (SHAP) methods. The permutation importance is applied based on the ELI5 python package, which can reflect the contribution of the predictors. In addition, the SHAP method is efficient to reflect the performance of each feature. Knowing the interpretability of the models is helpful for market investors.

2. Variables and methodology

2.1. Realized volatility (RV)

The monthly RV of the S&P 500 is expressed as follows:

$$\mathrm{RV}_t = \sum_{d=1}^{M_t} r_{t,j}^2 \tag{1}$$

where $r_{t,j}$ represents the d^{th} daily return of month t and M signifies the total count of return observations recorded during the t-th month.

2.2. Traditional predicting models

2.2.1. Autoregressive regression (AR) models

Following Paye (2012), the benchmark model (AR) model is:

$$\mathrm{Ln}(\mathrm{RV})_{t+1} = \alpha_0 + \sum_{k=1}^{K} \alpha_k \mathrm{Ln}(\mathrm{RV})_{t+1-k} + \varepsilon_{t+1} \tag{2}$$

where $\mathrm{Ln}(\mathrm{RV})_{t+1}$ is the natural logarithm of RV_{t+1}, α_k is the coefficient of k-lag Ln(RV), and ε_{t+1} is the disturbance term. We apply the AR(3) model as the benchmark model.

2.2.2. Principal component analysis (PCA) model

The AR model including PCA analysis method (AR-PCA) is:

$$\mathrm{Ln}(\mathrm{RV})_{t+1} = \alpha_0 + \sum\nolimits_{k=1}^{3} \alpha_k \mathrm{Ln}(\mathrm{RV})_{t+1-k} + \sum_{p=1}^{P} \delta_p F_{p,t}^{\mathrm{PCA}} + \varepsilon_{t+1} \tag{3}$$

where $F_{p,t}^{\mathrm{PCA}}$ is the principal component obtained from all the macroeconomic predictors.

2.2.3. Partial least squares (PLS) model

We extract the PLS factors in the following process. First, a time-series regression can be:

$$X_{n,t-1} = \alpha_{n,0} + \beta_n \mathrm{Ln}(\mathrm{RV})_t + e_{n,t-1} \tag{4}$$

Second, a cross-sectional regression can be:

$$X_{n,t} = \varphi_{n,0} + F_t^{\mathrm{PLS}} \widehat{\beta}_i + \mu_{n,t-1} \tag{5}$$

where F_t^{PLS}, is the PLS diffusion index[1].

Then, the AR-PLS model is expressed as follows:

$$\mathrm{Ln}(RV)_{t+1} = \alpha_0 + \sum\nolimits_{k=1}^{3} \alpha_k \mathrm{Ln}(\mathrm{RV})_{t+1-k} + \sum_{p=1}^{P} \delta_p F_t^{\mathrm{PLS}} + \varepsilon_{t+1} \tag{6}$$

2.3. Machine learning models

2.3.1. LASSO model

LASSO (Tibshirani, 1996) considers L_1 penalty term.

$$\widehat{\mathrm{Ln}(\mathrm{RV})}_{\mathrm{LASSO},t+1} = \widehat{\alpha}_{0,\mathrm{LASSO}} + \sum\nolimits_{k=1}^{3} \widehat{\alpha}_{k,\mathrm{LASSO}} \mathrm{Ln}(\mathrm{RV})_{t+1-k} + \sum_{n=1}^{N} \widehat{\beta}_{n,\mathrm{LASSO}} X_{n,t} \tag{7}$$

[1] For further information regarding the estimation, refer to (Zhang and Ma et al., 2019) and Zhang and Wei et al. (2019).

$$\widehat{\beta_{\text{LASSO}}} = \underset{\beta}{\text{argmin}}\left(\frac{1}{2(t-1)}\sum_{l=1}^{t-1}\left(\text{Ln}(\text{RV})_{l+1} - \alpha_0 - \sum_{k=1}^{3}\widehat{\alpha}_{k,\text{LASSO}}\text{Ln}(\text{RV})_{l+1-k} - \sum_{n=1}^{N}\beta_n X_{n,l}\right)^2 + \lambda\sum_{n=1}^{N}|\beta_n|\right) \tag{8}$$

where λ symbolizes the non-negative tuning parameter for the L_1 and α represents a strictly positive scalar valuea ($\alpha \in [0,1]$).

2.3.2. Ridge model

The ridge model of Hoerl and Kennard (1970) considers L_2 penalty function.

$$\widehat{\text{Ln}(\text{RV})}_{\text{ridge},t+1} = \widehat{\alpha}_{0,\text{ridge}} + \sum_{k=1}^{3}\widehat{\alpha}_{k,\text{ridge}}\text{Ln}(\text{RV})_{t+1-k} + \sum_{n=1}^{N}\widehat{\beta}_{n,\text{ridge}}X_{n,t} \tag{9}$$

$$\widehat{\beta_{\text{ridge}}} = \underset{\beta}{\text{argmin}}\left(\frac{1}{2(t-1)}\sum_{l=1}^{t-1}\left(\text{Ln}(\text{RV})_{l+1} - \alpha_0 - \sum_{k=1}^{3}\widehat{\alpha}_{k,\text{ridge}}\text{Ln}(\text{RV})_{l+1-k} - \sum_{n=1}^{N}\beta_n X_{n,l}\right)^2 + \lambda\sum_{n=1}^{N}\beta_n^2\right) \tag{10}$$

The meanings of the symbols λ and α correspond to their previous descriptions.

2.3.3. Elastic net (ENet) model

The elastic net (ENet) of Zou and Hastie (2005) accounts for penalties associated with both L_1 and L_2 norms. Zou and Hastie (2005), which can be expressed as:

$$\widehat{\text{Ln}(\text{RV})}_{ENet,t+1} = \widehat{\alpha}_{0,ENet} + \sum_{k=1}^{3}\widehat{\alpha}_{k,ENet}\text{Ln}(\text{RV})_{t+1-k} + \sum_{n=1}^{N}\widehat{\beta}_{n,ENet}X_{n,t} \tag{11}$$

where $\widehat{\beta}_n$ represents the reductive estimator of the regression coefficients. The $\widehat{\beta}_n$ of ENet is:

$$\widehat{\beta}_{\text{ENet}} = \underset{\beta}{\text{argmin}}\left(\frac{1}{2(t-1)}\sum_{l=1}^{t-1}\left(\text{Ln}(\text{RV})_{l+1} - \alpha_0 - \sum_{k=1}^{3}\widehat{\alpha}_{k,\hat{\beta}_{n,ENet}}\text{Ln}(\text{RV})_{l+1-k} - \sum_{n=1}^{N}\beta_n X_{n,l}\right)^2 + \lambda\sum_{n=1}^{N}\left((1-\alpha)\beta_n^2 + \alpha|\beta_n|\right)\right) \tag{12}$$

The meanings of the symbols λ and α correspond to their previous descriptions.

2.3.4. Adaptive LASSO (ALASSO)

The adaptive LASSO of Zou (2006) adds weight to the L_1 regularization term of LASSO and considers different weights to different coefficients, which can help correct the LASSO deviation. The weighted LASSO is expressed as:

$$\widehat{\text{Ln}(\text{RV})}_{\text{ALASSO},t+1} = \widehat{\alpha}_{0,\text{ALASSO}} + \sum_{k=1}^{3}\widehat{\alpha}_{k,\text{ALASSO}}\text{Ln}(\text{RV})_{t+1-k} + \sum_{n=1}^{N}\widehat{\beta}_{n,\text{ALASSO}}X_{n,t} \tag{13}$$

$$\widehat{\beta_{\text{ALASSO}}} = \underset{\beta}{\text{argmin}}\left\|y - \sum_{i=1}^{n}X_i\beta_i\right\|^2 + \lambda\sum_{i=1}^{n}w_i|\beta_i| \tag{14}$$

where y is the desired outcome, X is the input variables, β is the feature estimation coefficient, λ is the regularization term coefficient, and n is the number of features. w_i is the penalty weight of the ith feature, which can measure the features that are truly important. More specifically, the smaller the value, the more important the feature.

2.3.5. Adaptive elastic net (AEnet)

The adaptive elastic net model of Li et al. (2011) adds weight to the L_1 regularization term of the ENet method and considers different weights for different coefficients:

$$\widehat{\text{Ln}(\text{RV})}_{\text{AEnet},t+1} = \widehat{\alpha}_{0,\text{AEnet}} + \sum_{k=1}^{3}\widehat{\alpha}_{k,\text{AEnet}}\text{Ln}(\text{RV})_{t+1-k} + \sum_{n=1}^{N}\widehat{\beta}_{n,\text{AEnet}}X_{n,t} \tag{15}$$

$$\widehat{\beta_{\text{AEnet}}} = \left(1+\frac{\lambda_2}{n}\right)\underset{\beta}{\text{argmin}}\left\{L_n(\beta) + \lambda_2\sum_{i=1}^{n}\beta_i^2 + \lambda_1\sum_{i=1}^{n}w_i|\beta_i|\right\} \tag{16}$$

where y is the desired outcome, X is the input feature, β is the feature estimation coefficient, λ_1 and λ_2 are coefficients of L_1 and L_2 penalty functions, respectively, n is the number of features. w_i is the penalty weight of the ith feature, which can measure the features that are truly important. More specifically, the smaller the value, the more important the feature.

2.3.6. Group LASSO (GLASSO)

Yuan and Lin (2006) consider group features in LASSO methods and constructed a group LASSO method that can predict the target based on group features.

For a vector $\eta \in R^d$, $d \geq 1$, and a symmetric d by d positive definite matrix K, then

$$\|\eta\|_K = (\eta' K\eta)^{1/2} \tag{17}$$

Following Yuan and Lin (2006) Yuan and Lin (2006), $\|\eta\| = \|\eta\|_{I_d}$ for brevity, and group LASSO is as follows:

$$\widehat{\beta_{\text{GLASSO}}} = \underset{\beta}{\text{argmin}} 1/2 \left\| y - \sum_{j=1}^{J} G_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{J} \sqrt{p_j} \|\beta_j\| \tag{18}$$

where $\lambda \geq 0$ is the tuning parameter, j is the number of groups, y is the target value, G is the input group features, β_j is the feature estimation coefficient of group j and p_j is the length of β_j.

2.3.7. Adaptive group LASSO (AGLASSO)

Based on the group LASSO method, Wang and Leng (2008) add weight to the regularization term to measure the importance of different groups.

$$\widehat{\beta_{\text{AGLASSO}}} = \underset{\beta}{\text{argmin}} 1/2 \left\| y - \sum_{j=1}^{J} G_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{J} w_j \|\beta_j\| \tag{19}$$

where y is the desired outcome, G is the input group features, j signifies the count of distinct groups, λ represents the coefficient for the regularization term, β_j is the coefficient that estimates the features for the j-th group, and w_j denotes the weight assigned to the j-th group. Specifically, a lower value for w_j indicates a higher significance attributed to the features of that group.

2.3.8. Sparse group LASSO (SGLASSO)

Simon et al. (2013) extend group LASSO method by considering sparsity within different feature groups and adding it to the L_1 regularization term.

$$\widehat{\beta_{\text{SGLASSO}}} = \underset{\beta}{\text{argmin}} 1/2n \left\| y - \sum_{j=1}^{J} G_j \beta_j \right\|^2 + (1-\alpha)\lambda_g \sum_{j=1}^{J} \sqrt{p_j} \|\beta_j\| + \alpha \lambda_f \|\beta\|_1 \tag{20}$$

where y is the desired outcome, G is the input group features, j is the number of groups, λ_g is the coefficient of the regularization term of the group features, λ_f is the coefficient of the regularization term of all features, α is a convex combination of the LASSO and group LASSO penalties, and β_j is the feature estimation coefficient of group j.

2.3.9. Adaptive sparse group LASSO (ASGLASSO)

Fang et al. (2015) demonstrate that different groups and individual features play different roles in the prediction process and construct an adaptive sparse group LASSO method. More specifically, based on the sparse group LASSO model, two weight vectors are added to measure the importance of the group features and all individual features.

$$\widehat{\beta_{\text{ASGLASSO}}} = \underset{\beta}{\text{argmin}} 1/2n \left\| y - \sum_{j=1}^{J} G_j \beta_j \right\|^2 + (1-\alpha)\lambda_g \sum_{j=1}^{J} w_j \|\beta_j\| + \alpha \lambda_f \sum_{i=1}^{n} w_i |\beta_i| \tag{21}$$

where y is the desired outcome, G is the input group features, j is the number of groups, and β_j is the feature estimation coefficient of group j, λ_g is the coefficient of the regularization term of the group features, λ_f is the coefficient of the regularization term of all features, w_j is the penalty weight of the term group features, α is a convex combination of the LASSO and group LASSO penalties, and w_i is the penalty weight of the i-th feature.

2.3.10. Support vector regression (SVR) model

SVR aims to minimize the overall error by encompassing a greater number of sample points within a specified band of tolerance, which is determined by ε. The SVR can be:

$$y = wx + b \tag{22}$$

where $y = \text{Ln}(\text{RV})_{t+1}$, x presents a feature vector including the k-lags values of Ln(RV) and macroeconomic variables, w signifies the vector of weights assigned to these features, and b is the bias term.

When $C > 0$ and $\varepsilon > 0$, the standard SVR Vapnik (1998) is

$$\begin{cases} \underset{w,b,\S,\S^*}{min} \dfrac{1}{2} w^T w + C \sum_{i=1}^{k} \S_i + C \sum_{i=1}^{k} \S_i^* \\ \text{subject to } w^T x_i + b - y_i \leq \varepsilon + \S_i \\ \quad y_i - w^T x_i - b \leq \varepsilon + \S_i^* \\ \quad \S_i, \S_i^* \geq 0, i = 1, \ldots, k \end{cases} \tag{23}$$

where $\S_i$ and $\S_i^*$ denote the measurements of how far the support vectors extend beyond the band's edge, away from its boundary.

2.3.11. K-nearest neighbor (KNN) model

KNN (Altman, 1992) can find the k closest training samples and make predictions by considering the valuable content of the k 'neighbors'. The distance L_p is

$$L_p(x_i, x_j) = \left(\sum_{n=1}^{N} \left| x_i^N, x_j^N \right|^p \right)^{\frac{1}{p}} \tag{24}$$

where $x_i = (x_i^1, x_i^2, \ldots, x_i^N)$ and $x_j = \left(x_j^1, x_j^2, \ldots, x_j^N\right)$ represent the vectors of attributes for the data instances, $p \geq 1$.

2.3.12. Decision trees (DT) model

DT (Loh, 2011) is a nonparametric method. The least squares error criterion was used to select the features and generate a binary tree. Lnitially, identify the most suitable variable v for partitioning and the corresponding divisionpoint s. Subsequently, examine the variable u to locate the division point s for a specifiedpartitioning variable v and choose the (v, s) that minimizes Eq. (25).

$$\underset{v,s}{min} \left[\underset{c_1}{min} \sum_{x_i \in R_1(v,s)} (y_i - c_1)^2 + \underset{c_2}{min} \sum_{x_i \in R_2(v,s)} (y_i - c_2)^2 \right] \tag{25}$$

where x_i and y_i denote the feature and response variables of the training dataset, respectively. R_d refers to the subset of the input space that is being segmented, while C_d isthe resultant value associated with the segmented region R_d.

Second, we segment the area., which is:

$$\begin{cases} R_1(v,s) = \{x | x^v \leq s\} \\ R_2(v,s) = \{x | x^v > s\} \\ \widehat{c_d} = \dfrac{1}{N_d} \sum_{x_i \in R_d(v,s)} y_i \\ x \in R_d, d = 1, 2 \end{cases} \tag{26}$$

where N_d signifies the count of data points within the domain R_d.

Third, we proceed to iterate through steps one and two until the

termination criterion is satisfied.

Finally, we segment the input domain into D distinct sectors (R_1, R_2, …, R_d), and construct a decision tree.

$$f(x) = \sum_{d=1}^{D} \hat{c}_d I(x \in R) \tag{27}$$

2.3.13. Extra trees (ET) model

ET (Geurts et al., 2006) is similar to random forecasts but has differences.

(1) ET algorithm employs the original training dataset for the construction of each individual decision tree.
(2) ET randomly picks a feature for partitioning each node of the decision tree.

By randomly selecting the division point of feature values, the ET system constructs a decision tree that is larger in scope than a random prediction, which in turn yields a lower variance but a higher deviation for ET than for the random forecast.

2.3.14. Random forest (RF) model

RF (Breiman, 2001) incorporates a stochastic selection of features during the training phase.

First, the bootstrapping technique was utilized to create k distinct training datasets by sampling with replacement from the original dataset. Following this, a decision tree was constructed for each set.

Second, a random subset of N' features was chosen, the best split was determined, and the node was divided based on this criterion.

Finally, the decision tree construction was continued until no additional splits could be made.

RF models are proficient in delivering low variance, minimal bias, and robust generalization performance across various datasets.

2.3.15. Bagging model

Bagging (Bauer & Kohavi, 1999) is a quintessential parallel ensemble technique that leverages bootstrap sampling. Starting with a dataset of N instances, the method involves randomly selecting a sample for inclusion in the training set, followed by returning it to the pool, thus allowing for its potential re-selection in subsequent iterations. This bagging approach employs a parallel averaging mechanism during training, which is underpinned by random sampling, and subsequently amalgamates the individual learners into a robust ensemble, thereby diminishing the variance inherent in the models.

2.3.16. Adaboost model

Adaboost (Freund & Schapire, 1996) undergoes multiple training cycles. The weights of the samples are changed to identify the optimal weak classifier under the prevailing distribution, and the error rate of this classifier is determined. The sample weights are then revised based on the error rate, with all samples beginning with equal weights. In the training process, samples that are misclassified by the weak classifier receive increased weight, thereby drawing more attention in subsequent training rounds, and samples correctly classified have their weights reduced. The accuracy of the weak classifier directly influences its weight, with higher accuracy classifiers being assigned greater influence. The aggregation of these weak classifiers, trained over several iterations, results in the formation of a strong classifier, with a lower prediction error signifying a more significant role in the ensemble.

2.3.17. Gradient boosting regression (GBR) model

GBR (Friedman, 2001) distinguishes itself from the Adaboost model by choosing the gradient descent direction during each iteration to optimize the final outcome. The core concept of GBR is the sequential creation of multiple weak learners. Each individual weak predictor in GBR is designed to align with the negative slope of the loss function relative to the current model, with the goal of diminishing the overall model's loss in the direction of this gradient upon its inclusion. The final prediction of the Gradient Boosting Regression (GBR) model is obtained by synthesizing and weighing the predictions from each constituent weak learner across the training iterations.

2.3.18. XGBoost model

XGBoost (Chen & Guestrin, 2016) is an engineered implementation of the GBDT (Gradient Boosting Decision Tree) algorithm. Similar to GBDT, XGBoost is an additive model, and its core is to adopt the idea of boosting to integrate multiple weak learners into strong learners through certain methods. In each iteration, XGBoost optimises only the weak learner in the current step, which learns the difference between the predicted result of the previously trained weak base learner and the actual value. By learning multiple weak learners, XGBoost can continuously reduce the difference. In contrast to the GBDT, XGBoost adds a regularization term to control the complexity of the model, which can effectively prevent overfitting. Furthermore, GBDT employs the first derivative of the loss function, akin to gradient descent within the functional domain. XGBoost, on the other hand, utilizes both the first and second derivatives of the loss function, which is analogous to Newton's method in the context of the functional space.

2.3.19. Fully connected neural networks (FCNN) model

The FCNN (Lewenstein & Nowak, 1989) is composed of an input layer, followed by one or more intermediate hidden layers, culminating in an output layer.

Within these layers, neurons operate independently of each other, and each neuron in the subsequent layer is connected to every neuron in the preceding layer. The FCNN boasts an impressive fitting capacity due to the adaptability of neural networks to accommodate a wide array of functions.

2.3.20. Convolutional neural networks (CNN) model

A CNN (LeCun et al., 1989) represents a class of feedforward artificial neural networks where artificial neurons are sensitive to their surrounding input units. CNN feature convolutional layers that utilize filters to process the input data, coupled with pooling layers designed to downscale the spatial extent of the feature maps, enhancing the network's ability to process image data.

The formula is:

$$a^k = f\left(w^k x^{k-1} + b^k\right) \tag{28}$$

where k denotes the k-th layer in the sequence, x^{k-1} represents the output from the preceding layer k-1. w^k signifies the set of convolutional filters associated with the transitionfrom layer k-1 to layer k, b^k is the bias term for layer k, and a^k is the resultant output fromlayer k.

The rectified linear unit (ReLU) is frequently chosen as an activation function that introduces nonlinearity. It allows the network to deactivate the output of specific neurons, which promotes sparsity and helps to reduce the likelihood of overfitting.

$$a^k = f\left(y^k\right) = max\left(0, y^k\right) \tag{29}$$

where y^k signifies the result yielded by the convolution process, while a^k denotes theactivated output corresponding to y^k.

2.3.21. Recurrent neural network (RNN)

An RNN (HIHI, E. L., 1996) is a time-dependent network that specializes in processing sequence information. In an RNN, each output is determined by both the current input and the previous information. The RNN is calculated as follows:

$$s_t = f(u \bullet x_t + w \bullet s_{t-1}) \tag{30}$$

$$o_t = g(v \bullet s_t) \tag{31}$$

where x_t represents the input at the present time step, s_t is the state of the RNN's hidden layers at the current time step, s_{t-1} denotes the state of the hidden layers at the preceding time step, o_t is the output at the present time step, u aptures the weights from the input to the hidden layers, v captures the weights from the input to the hidden layers, w refers to the weight associated with the previous state s_{t-1}, $f(\bullet)$ and $g(\bullet)$ refers to the weight associated with the previous state. It can be seen from the formula that the value of s_t depends not only on x_t, but also on s_{t-1}.

However, when the input sequence becomes longer, the RNN may face the problems of gradient disappearance and gradient explosion. The gradient disappearance reduces the influence of the previous data on the later prediction, leading to a loss of the ability to learn the previous information.

2.3.22. Long short-term memory (LSTM) model

LSTM (Schmidhuber, 1992) was specifically designed to mitigate the issues associated with the traditional RNN model, particularly the challenges of gradient explosion and vanishing gradients during the training of extended sequences.

The formulas are:

$$\begin{cases} f_t = \sigma(w_f \bullet [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(w_i \bullet [h_{t-1}, x_t] + b_i) \\ \widetilde{c_t} = tanh(w_c \bullet [h_{t-1}, x_t] + b_c) \\ c_t = f_t c_{t-1} + i_t \widetilde{c_t} \\ o_t = \sigma(w_o \bullet [h_{t-1}, x_t] + b_o) \\ h_t = o_t tanh(c_t) \end{cases} \tag{32}$$

where h_{t-1} enotes the preceding layer's output, x_t is the input for the current timestep, σ acts as the sigmoid function that confines values to the interval [0,1], w representsthe weights of the neural connections, b is the bias within the network, c_{t-1} refers to the storedinformation from the previous instant, c_t is the current memory state, $\widetilde{c_t}$ is the provisionalmemory for the ongoing LSTM phase, and h_t is the LSTM cell's resultant output.

2.3.23. C_LSTM model

The Convolutional Long Short-Term Memory (C-LSTM) model integrates the swift and efficient processing capabilities of the CNN with the temporal sequence sensitivity of the LSTM network. Initially, the C-LSTM employs a CNN layer to create various filters designed to mitigate noise and capture brief high-level features. Subsequently, these high-level features are funneled into the LSTM layer to distill the temporal dynamics of the data.

2.3.24. RF-LASSO model

The random forest model exhibits good anti-noise ability by introducing randomness in the process, and it also applies unbiased estimation for generalization errors. However, some studies have confirmed that the random forest model has overfitting issues when noisy classification or regression problems exist (Cutler et al., 2012; Hastie et al., 2009; Tang et al., 2018). The LASSO method is well known for dealing with overfitting issues (Lu et al., 2021; Ma et al., 2023; Zhang, Wei, et al., 2019). Thus, we comprehensively examine the advantages and disadvantages of the random forest and LASSO methods and construct a new hybrid model, named the RF-LASSO model. The process consists of the following three steps.

(1) feature importance calculation.

The random forest model uses the Gini coefficient for feature division so that the child nodes of the established tree model have less uncertain information than the parent nodes. In the process of building the tree model, the noise information is gradually reduced and the contribution of each feature is calculated.

Consider N variables $\{x_1, x_2, \ldots, x_{n-1}, x_n\}$ and M binary-tree models. Subsequently, the Gini coefficient of each tree model at node f is calculated as follows:

$$GI_f = \sum_{k=0}^{1} \sum_{k=1} p_{f0} p_{f1} \tag{33}$$

where p_{f0} represents the proportion of node f in class 0 samples and p_{f1} represents the proportion of node f in class 1 samples.

Then, the importance (I_i) of x_i in node f can be expressed by the change in the Gini coefficient before and after the branching of node f:

$$I_{if} = GI_f - GI_0 - GI_1 \tag{34}$$

where GI_0 and GI_1 are the Gini coefficients of the left and right child nodes of node f, respectively, after branching.

$$I_{if} = GI_f - GI_0 - GI_1 \tag{35}$$

The importance of x_i in the binary tree m is:

$$I_{im} = \sum_{f \in F} I_{if} \tag{36}$$

where F is the set of nodes where variable x_i appears in the binary tree m.

Subsequently, the importance of x_i in all the binary trees is:

$$I_i = \sum_{m=1}^{M} I_{im} \tag{37}$$

Finally, the importance scores of all variables were normalized to obtain the final importance score, I_i'. Accordingly, the importance of x_i is sorted in descending order to obtain the importance sequence, s_i.

$$I_i' = \frac{I_i}{\sum_{i=1}^{N} I_i} \tag{38}$$

In this study, the importance of variables I_i'' was calculated for variable selection. We handle the importance score I_i' using sparse processing.

$$I_i'' = \begin{cases} 0, if\ s_i > \sigma^* M \\ 1, if\ s_i \le \sigma^* M \end{cases} \tag{39}$$

where $\sigma \in [0, 1]$ is a manually determined retention threshold and σ is 0.8 in this paper.

(2) Variable coefficients

By combining I_i'' and LASSO algorithms, the selection coefficient $\widehat{\beta}_i$ of variable x_i is estimated as follows:

$$\widehat{\beta}_i = \underset{\beta}{\operatorname{argmin}} \left(\frac{1}{2(t-1)} \sum_{l=1}^{t-1} \left(\mathrm{Ln}(\mathrm{RV})_{l+1} - \sum_{i=1}^{N} \beta_i I_i'' X_{i,l} \right)^2 + \lambda \sum_{i=1}^{N} |\beta_i| \right) \tag{40}$$

where λ symbolizes the positive scaling factor for the L_1 regularization term.

(3) Predictions

LASSO is a biased estimator and the obtained coefficient is different from the optimal coefficient. Therefore, this study applies the improved hybrid model for variable selection, and finally obtains the unbiased estimated coefficient $\widehat{\alpha}$ using the least squares method.

$$\widehat{\mathrm{Ln}(\mathrm{RV})}_{t+1} = \widehat{\alpha}_0 + \sum_{k=1}^{3} \widehat{\alpha}_k \mathrm{Ln}(\mathrm{RV})_{t+1-k} + \sum_{i=1}^{N} \widehat{\beta}_i \widehat{\alpha}_i X_{i,t} \tag{41}$$

Table 1
Out-of-sample results.

Forecasting models	R^2_{OOS}(%)	MSPE-Adj.	*p*-value	Utility
AR				1.4237
PCA	−1.9760	3.1572	0.0008	1.5938
PLS	0.8459	2.8969	0.0019	1.6725
LASSO	**10.0494**	4.7122	0.0000	1.7616
Ridge	**10.0494**	4.7122	0.0000	1.7616
ENet	**10.0494**	4.7122	0.0000	1.7616
ALASSO	**10.3241**	4.7853	0.0000	1.8547
AENet	**10.3171**	4.7860	0.0000	1.8546
GLASSO	**11.7850**	4.9319	0.0000	1.8898
AGLASSO	**10.6076**	4.8560	0.0000	1.8592
SGLASSO	**13.2728**	5.2074	0.0000	1.9087
ASGLASSO	**10.5490**	4.8376	0.0000	1.8588
SVR	−9.4776	3.2016	0.0007	1.0894
KNN	−9.0037	2.7657	0.0028	1.4162
ET	**9.1416**	5.0168	0.0000	2.1068
DT	**9.7695**	5.1725	0.0000	1.8437
RF	**27.8181**	6.0064	0.0000	2.3841
Bagging	**26.3208**	5.7595	0.0000	2.2736
Adaboost	**25.8291**	5.3576	0.0000	2.4537
GBR	−10.1032	2.4743	0.0067	1.4837
XGB	**20.0692**	5.4781	0.0000	2.2858
FCNN	**20.0345**	4.1885	0.0000	1.9918
CNN	**5.0122**	5.0663	0.0000	1.9765
RNN	−2.7078	4.8147	0.0000	1.9496
LSTM	−1.4525	5.0916	0.0000	1.5606
C_LSTM	**15.4874**	4.4996	0.0000	2.1218
RF-LASSO	**29.2817**	5.0870	0.0000	2.3409

Notes: The table displays the post-sample assessment metrics for model performance, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

3. Data

In our research, we delve into the fluctuations of the stock market by examining more than a hundred monthly indicators from the FRED-MD database, as reported by McCracken and Ng (2016) McCracken and Ng (2016).[2] TThe economic indicators within FRED-MD are categorized into eight distinct sectors: output and income, labour market, housing, consumption, orders, inventories, money and credit, interest and exchange rates, prices, and stock market. To account for past data, we utilize 126 economic indicators for forecasting stock market volatility, with our data ranging from March 1978 to October 2021. Further statistical details can be found in the supplementary material online.

4. Empirical analysis

4.1. Out-of-sample analysis

4.1.1. Results based on out-of-sample R^2 test

The out-of-sample R^2 statistic (R^2_{OOS}) is

$$R^2_{OOS} = 1 - \frac{\sum_{t=1}^{M}\left(\mathrm{Log}(\mathrm{RV}_t) - \mathrm{Log}(\mathrm{RV}_t^j)\right)^2}{\sum_{t=1}^{M}\left(\mathrm{Log}(\mathrm{RV}_t) - \mathrm{Log}(\mathrm{RV}_t^0)\right)^2} \tag{42}$$

where $\mathrm{Log}(\mathrm{RV}_t)$, $\mathrm{Log}\left(\mathrm{RV}_t^j\right)$ and $\mathrm{Log}\left(\mathrm{RV}_t^0\right)$ represent the actual realized volatility, prediction from model *j*, and the benchmark model respectively. A model with a positive R^2_{OOS}, it indicates that it has better forecasting performance. Following Clark and West (2007), the MSPE metric is used to examine the differences.

The out-of-sample period ranges from January 2000 to October 2021. Table 1 presents the results. First, we find that 20 out of 26 models have significantly positive R^2_{OOS} values; for instance, the regularization models, ensemble learning models, and neural network models show that most of the models have better forecasting performance. Second, we find that the R^2_{OOS} value of random forest is 27.8181, which is the largest among the models. This result shows that random forest has a relatively better forecasting performance than commonly applied machine learning models. Third, the new hybrid model we constructed by combining the random forest and LASSO methods has a R^2_{OOS} value of 29.2817, showing that this new hybrid model can further improve the forecasting accuracy of random forest and perform best. Possible reason can be that the combination of the random forest and LASSO methods can better learn from each other, make up for the lack of random forest. Thus, the RF-LASSO model has relatively better predictive performance for the stock market volatility.

4.1.2. Economic value analysis

According to Bollerslev et al. (2018), the expected utility function is:

$$E_t(u(W_{t+1})) = E_t(W_{t+1}) - \frac{1}{2}\gamma \mathrm{Var}_t(W_{t+1}) \tag{43}$$

where W_{t+1} signifies the distribution of investments at the subsequent time point $t+1$, and γ is the coefficient reflecting the investor's aversion to risk. Assuming w_t^* is the optimal weight, with the balance being invested in a risk-free asset. $W_{t+1} = W_t\left(1 + w_t^* r_{t+1} + \left(1 - w_t^*\right) r_{t+1f}\right) = W_t\left(1 + r_{t+1f}\right) + W_t\left(w_t^* \widehat{r_{t+1}}\right)$, where $\widehat{r_{t+1}}$ is the excess return ($\widehat{r_{t+1}} = r_{t+1} - r_{t+1f}$), r_{t+1} is the stock return, and r_{t+1f} is the risk-free rate. The optimal proportion is.

$$w_t^* = \frac{1}{\gamma}\left(\frac{E_t(\widehat{r_{t+1}})}{E_t(\mathrm{RV}_{t+1})}\right) \tag{44}$$

Table 2
Alternative stock index.

Forecasting models	R^2_{OOS}(%)	MSPE-Adj.	*p*-value
PCA	−3.3557	2.2934	0.0109
PLS	−2.2196	2.0976	0.0180
LASSO	**17.6798**	7.4440	0.0000
Ridge	**18.2580**	7.4788	0.0000
ENet	**10.4682**	6.3348	0.0000
ALASSO	**18.2872**	7.4799	0.0000
AENet	**18.2506**	7.4746	0.0000
GLASSO	**17.5940**	7.2490	0.0000
AGLASSO	**18.3751**	7.5050	0.0000
SGLASSO	**17.6139**	7.2474	0.0000
ASGLASSO	**18.2300**	7.4732	0.0000
SVR	−0.0340	5.2153	0.0000
KNN	−1.1413	4.3127	0.0000
ET	**10.5418**	6.4360	0.0000
DT	**15.6659**	7.1918	0.0000
RF	**31.9934**	8.4372	0.0000
Bagging	**30.7470**	8.2806	0.0000
Adaboost	**30.1839**	8.1229	0.0000
GBR	**0.7376**	4.4811	0.0000
XGB	**24.5434**	7.9652	0.0000
FCNN	**27.4569**	6.3751	0.0000
CNN	**7.8330**	6.7218	0.0000
RNN	**0.1821**	5.8828	0.0000
LSTM	**4.9354**	5.7211	0.0000
C_LSTM	**24.9035**	7.3704	0.0000
RF-LASSO	**35.0893**	7.5254	0.0000

Notes: The table displays the post-sample assessment metrics for model performance, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

[2] More details regarding the macroeconomic variables can refer to https://research.stlouisfed.org/econ/mccracken/sel/.

Table 3
Different forecasting method.

Forecasting models	R^2_{OOS}(%)	MSPE_adjusted	p_value
PCA	−15.1485	3.0479	0.0012
PLS	**0.2599**	2.7941	0.0026
LASSO	**11.2432**	5.4973	0.0000
Ridge	**11.5629**	5.3895	0.0000
ENet	**0.6113**	4.3854	0.0000
ALASSO	**11.5629**	5.3895	0.0000
AENet	**11.5629**	5.3895	0.0000
GLASSO	**10.7335**	5.2915	0.0000
AGLASSO	**10.0240**	5.1152	0.0000
SGLASSO	**10.8021**	5.3198	0.0000
ASGLASSO	**10.0777**	5.1146	0.0000
SVR	−9.3384	3.4979	0.0002
KNN	−9.0482	3.1476	0.0008
ET	−1.3690	4.1091	0.0000
DT	−13.4771	5.1394	0.0000
RF	**29.2727**	5.8970	0.0000
Bagging	**25.3547**	5.6838	0.0000
Adaboost	**26.4874**	5.6058	0.0000
GBR	−9.6827	2.8001	0.0026
XGB	**23.4885**	5.8194	0.0000
FNN	**20.8823**	4.1530	0.0000
CNN	**4.6697**	5.5556	0.0000
RNN	−4.8430	4.3948	0.0000
LSTM	−4.1452	4.8104	0.0000
C_LSTM	**13.9462**	5.2674	0.0000
RF-LASSO	**29.5669**	5.1454	0.0000

Notes: The table displays the post-sample assessment metrics for model performance, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

Table 4
Different forecasting windows.

Forecasting models	R^2_{OOS}(%)	MSPE-Adj.	p-value	R^2_{OOS}(%)	MSPE-Adj.	p-value
	Panel A: 1/3 out-of-sample			Panel B: 2/3 out-of-sample		
PCA	−2.2975	3.2301	0.0006	−1.5386	3.2161	0.0006
PLS	**2.8935**	3.4463	0.0003	**0.6943**	3.2011	0.0007
LASSO	**12.2208**	4.3575	0.0000	**7.8297**	5.5865	0.0000
Ridge	**12.0540**	4.3953	0.0000	**7.7506**	5.5489	0.0000
ENet	**11.2317**	4.3919	0.0000	**5.3410**	5.5739	0.0000
ALASSO	**12.0482**	4.3933	0.0000	**7.5560**	5.5914	0.0000
AENet	**12.0519**	4.3949	0.0000	**7.6331**	5.5603	0.0000
GLASSO	**13.3725**	4.4899	0.0000	**8.5472**	5.7164	0.0000
AGLASSO	**12.1713**	4.4144	0.0000	**5.2716**	5.2289	0.0000
SGLASSO	**14.8888**	4.7399	0.0000	**9.2330**	5.8534	0.0000
ASGLASSO	**12.0994**	4.3945	0.0000	**5.2230**	5.2132	0.0000
SVR	−5.4563	3.2485	0.0006	−10.2726	3.3497	0.0004
KNN	−2.3978	3.0687	0.0011	−12.6620	2.4347	0.0075
ET	**6.7677**	4.2846	0.0000	**11.3609**	5.8114	0.0000
DT	**11.5700**	4.1612	0.0000	**2.8338**	5.2158	0.0000
RF	**26.9255**	5.0149	0.0000	**26.5189**	6.7047	0.0000
Bagging	**26.4696**	4.9505	0.0000	**24.9295**	6.4221	0.0000
Adaboost	**24.9498**	4.4651	0.0000	**26.1731**	6.1966	0.0000
GBR	−2.7046	2.9412	0.0016	−14.3781	1.9696	0.0244
XGB	**20.2890**	4.6731	0.0000	**18.7301**	6.2152	0.0000
FCNN	**20.7228**	3.3902	0.0003	**17.7931**	4.0791	0.0000
CNN	**14.1804**	4.0717	0.0000	**3.2550**	4.3361	0.0000
RNN	**4.7611**	4.1510	0.0000	**4.9242**	6.1188	0.0000
LSTM	**0.3401**	3.2217	0.0006	−2.2596	4.6473	0.0000
C_LSTM	**16.4993**	4.0293	0.0000	**13.8431**	5.0745	0.0000
RF-LASSO	**28.2781**	4.1281	0.0000	**28.5560**	5.7469	0.0000

Notes: The table displays the post-sample assessment metrics for model performance, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

where $\hat{r}_{t+1}$ is utilized to represent the predicted excess returns. Beyond that, $\widehat{\mathrm{RV}_{t+1}}$ is the volatility prediction. Furthermore, the conditional Sharpe ratio is also considered in the analysis, defined as $\mathrm{SR} = E_t(\widehat{r_{t+1}})/\sqrt{E_t(\mathrm{RV}_{t+1})}$, the expected utility of the portfolio can be expressed as:

$$\begin{aligned} U(\omega_t) &= W_t\left(w_t^* E_t(\widehat{r_{t+1}}) - \frac{\gamma}{2}\omega_t^{*2}\mathrm{Var}(\widehat{r_{t+1}})\right) \\ &= W_t\left(w_t^* \mathrm{SR}\sqrt{E_t(\mathrm{RV}_{t+1})} - \frac{\gamma}{2}\omega_t^{*2}E_t(\mathrm{RV}_{t+1})\right) \end{aligned} \tag{45}$$

Thus, the expected utility can be expressed as:

$$U(\omega_t^*) = \frac{SR^2}{2\gamma}W_t = \frac{1}{2}\times SR \times \frac{SR}{\gamma}W_t \tag{46}$$

However, $E_t(\mathrm{RV}_{t+1})$ cannot be obtained. The expected utility can be expressed as:

$$\mathrm{U}(\widehat{\mathrm{RV}}_{t+1}) = \frac{\mathrm{SR}^2}{\gamma}\left(\frac{\sqrt{\mathrm{RV}_{t+1}}}{\sqrt{\widehat{\mathrm{RV}}_{t+1}}} - \frac{1}{2}\frac{\mathrm{RV}_{t+1}}{\widehat{\mathrm{RV}}_{t+1}}\right) \tag{47}$$

$$\overline{U}(\widehat{\mathrm{RV}}_{t+1}) = \frac{1}{q}\sum_{t=m+1}^{m+q}\frac{SR^2}{\gamma}\left(\frac{\sqrt{\mathrm{RV}_{t+1}}}{\sqrt{\widehat{\mathrm{RV}}_{t+1}}} - \frac{1}{2}\frac{\mathrm{RV}_{t+1}}{\widehat{\mathrm{RV}}_{t+1}}\right) \tag{48}$$

Following Bollerslev et al. (2018), the annual SR is set to 0.4, and γ is set to 2. The results are presented in Table 1. The top three models with higher utility are the Adaboost, RF, and RF-LASSO models, showing that ensemble learning and the new hybrid models also have economic performance. It worth mentioning that the RF-LASSO model not only demonstrates statistical significance but also yields substantial economic value.

4.2. Various robustness tests

4.2.1. Alternative stock index

In this section, we consider an alternative stock index, Dow Jones Industrial Average (DJIA) index, in the United States (U.S.). We further explore whether the models help predict DJIA volatility. Table 2 presents the results. We find that the R^2_{OOS} values of 22 out of the 26 models are significantly positive, showing that most models have good performances. The R^2_{OOS} values of RF and RF-LASSO are 31.9934 % and 35.0893 %, respectively, which exceed those of alternative models, showing the integration of Random Forest with the LASSO technique offers superior efficacy.

4.2.2. Alternative forecasting method

Table 3 presents the results based on rolling window method. First, the R^2_{OOS} values for 18 out of the 26 models are significantly larger than zero, indicating that most of the models can better perform. In addition, based on the popular machine learning model, random forest has a R^2_{OOS} value of 29.2727, which is the largest and shows random forest has a relatively superior performance. Importantly, the R^2_{OOS} value of RF-LASSO is 29.5669, indicating that the combination of random forest and LASSO methods has the best forecasting performance.

4.2.3. Results based on different forecasting windows

Table 4 presents the results of different forecasting windows (1/3 and 2/3 out-of-sample periods). First, we find that the random forest model has a relatively better performance than commonly applied models. Second, the R^2_{OOS} values of RF-LASSO based on 1/3 and 2/3 out-of-sample windows are 28.2781 and 28.5560, respectively, showing RF-LASSO model can increase the predictability of random forest model.

4.3. Model explanation

Focusing on model interpretability is important in the machine learning field. In this paper, we try to explain how the model works and which variables are important for stock market volatility from different

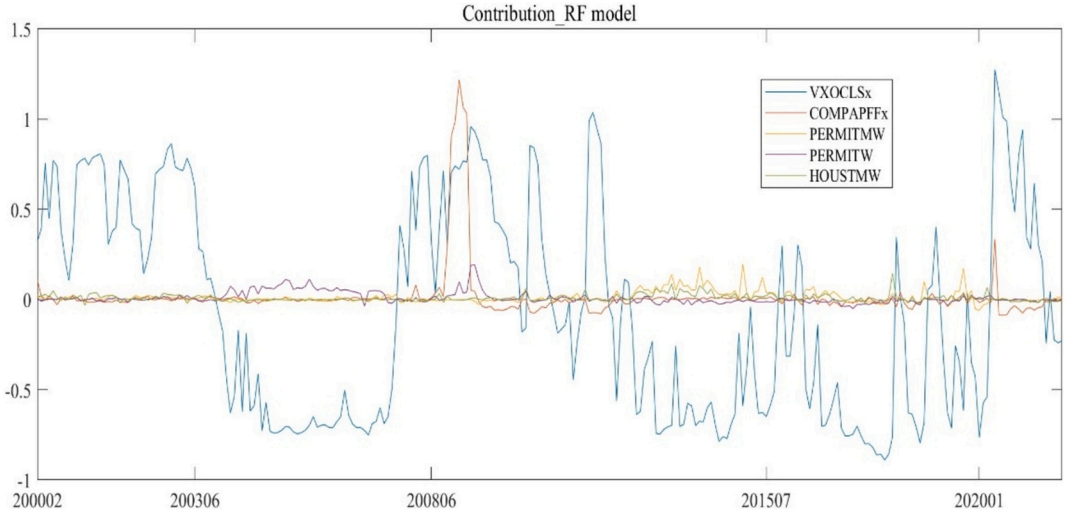

Fig. 1. This figure shows the contribution change of the top 5 features in the RF model.

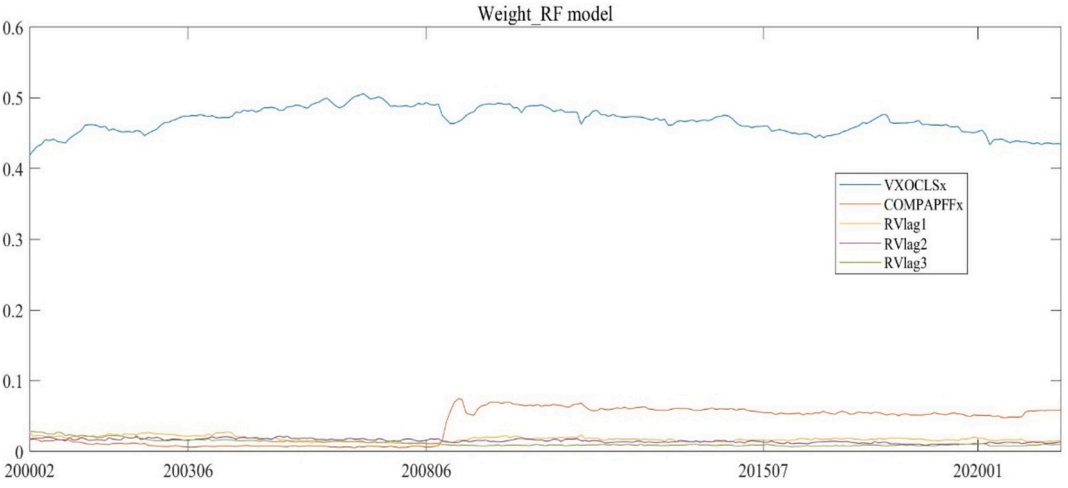

Fig. 2. This figure shows the weight change of the top 5 features in the RF model.

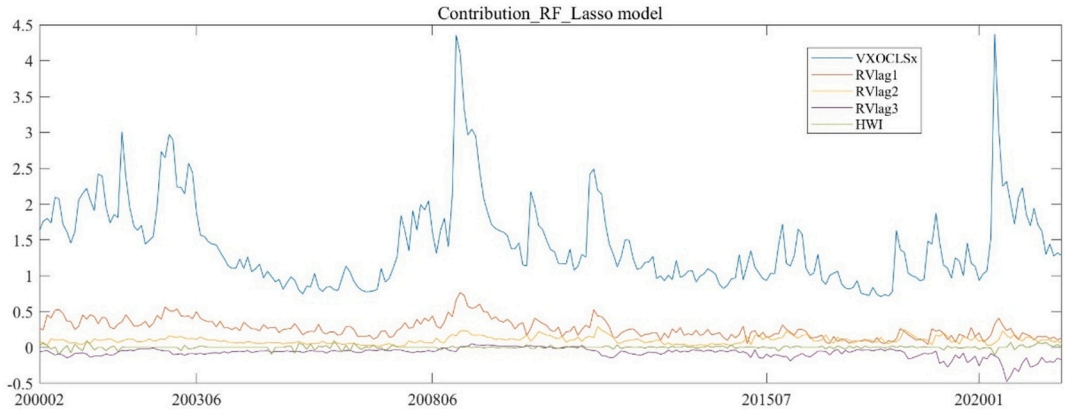

Fig. 3. This figure shows the contribution change of the top 5 features in the RF-LASSO model.

perspectives, such as permutation importance and Shapley additive explanation methods.

4.3.1. Permutation importance

It is interesting to determine how much this feature matters for prediction, which can be calculated using permutation importance based on the ELI5 python package. More specifically, if the arrangement of one feature was changed, and the remaining features remained unchanged, we can then see how much impact it has on prediction accuracy. Owing to space limitations, we only focus on two models with

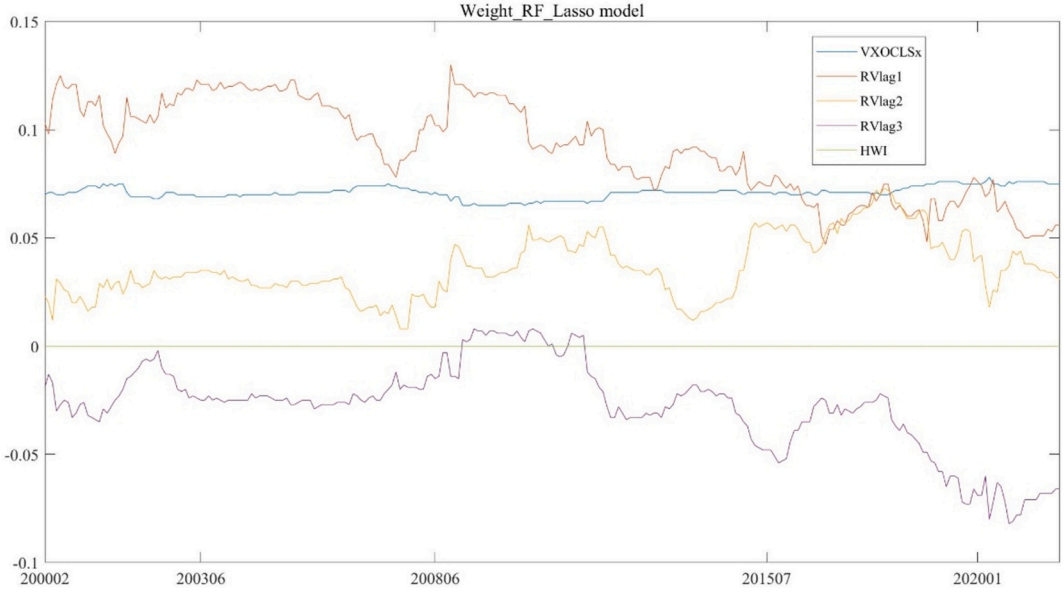

Fig. 4. This figure shows the weight change of the top 5 features in the RF-LASSO model.

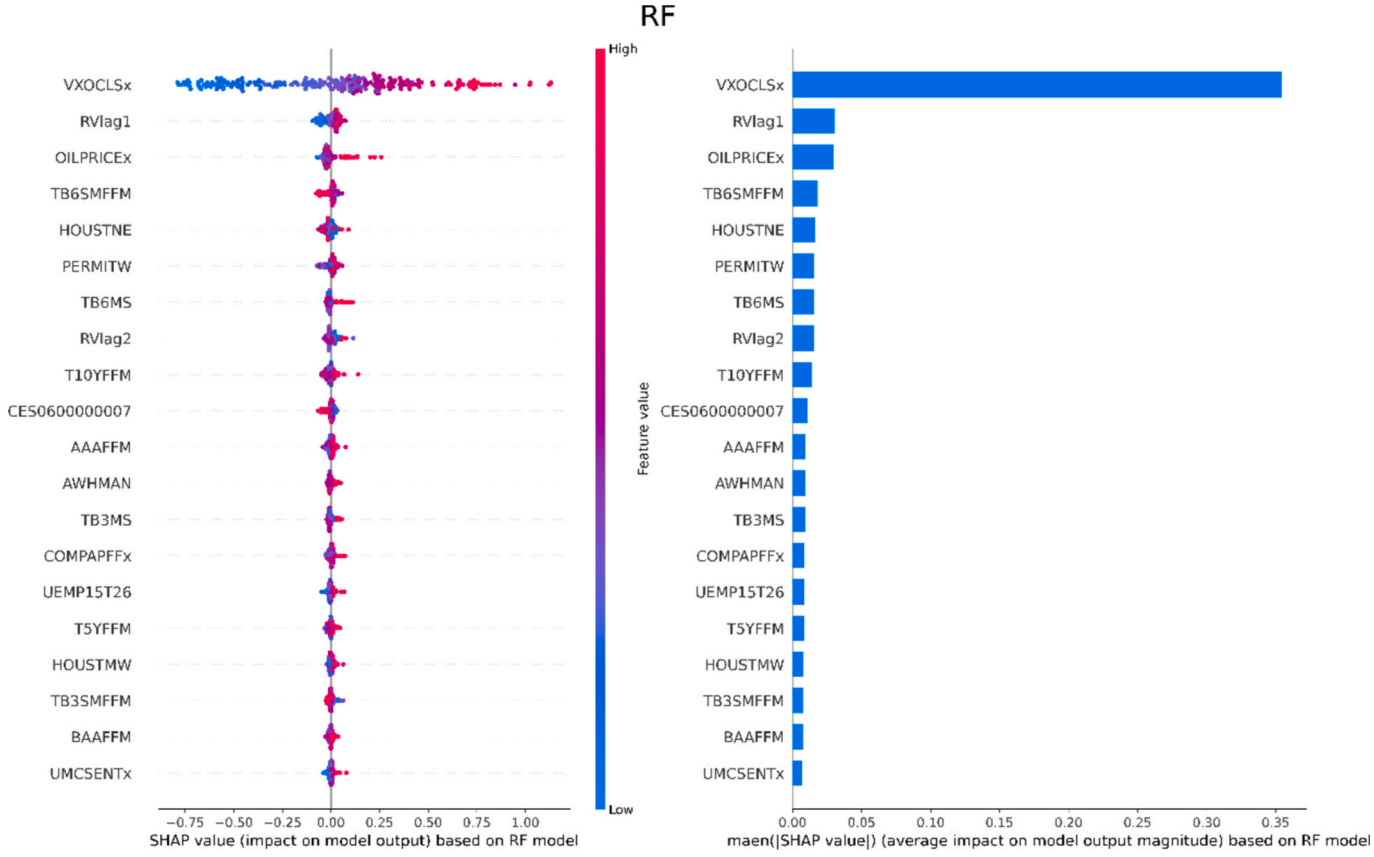

Fig. 5. This figure shows the model interpretability based on the SHAP method of RF model.

relatively better performance. Figs. 1 and 2 show the contributions and weights of the top five features during the out-of-sample period in the RF model. Similar information for the RF-LASSO model can be found in Figs. 3 and 4. From Figs. 1 to 4, we find that the VXOCLSx predictor plays the most important role among all the predictors of stock market volatility. McCracken and Ng (2016) construct the VXOCLSx predictor by splicing historical VXO data with the VIX, providing a measure of stock market volatility. We get an important finding that, among the macroeconomic variables, the VXOCLSx predictor contributes the most to predicting stock market volatility.

4.3.2. *Shapley additive explanation method*

The SHAP method considers the contributors of all features and explains the output of the model, which is developed in Python. Denoting

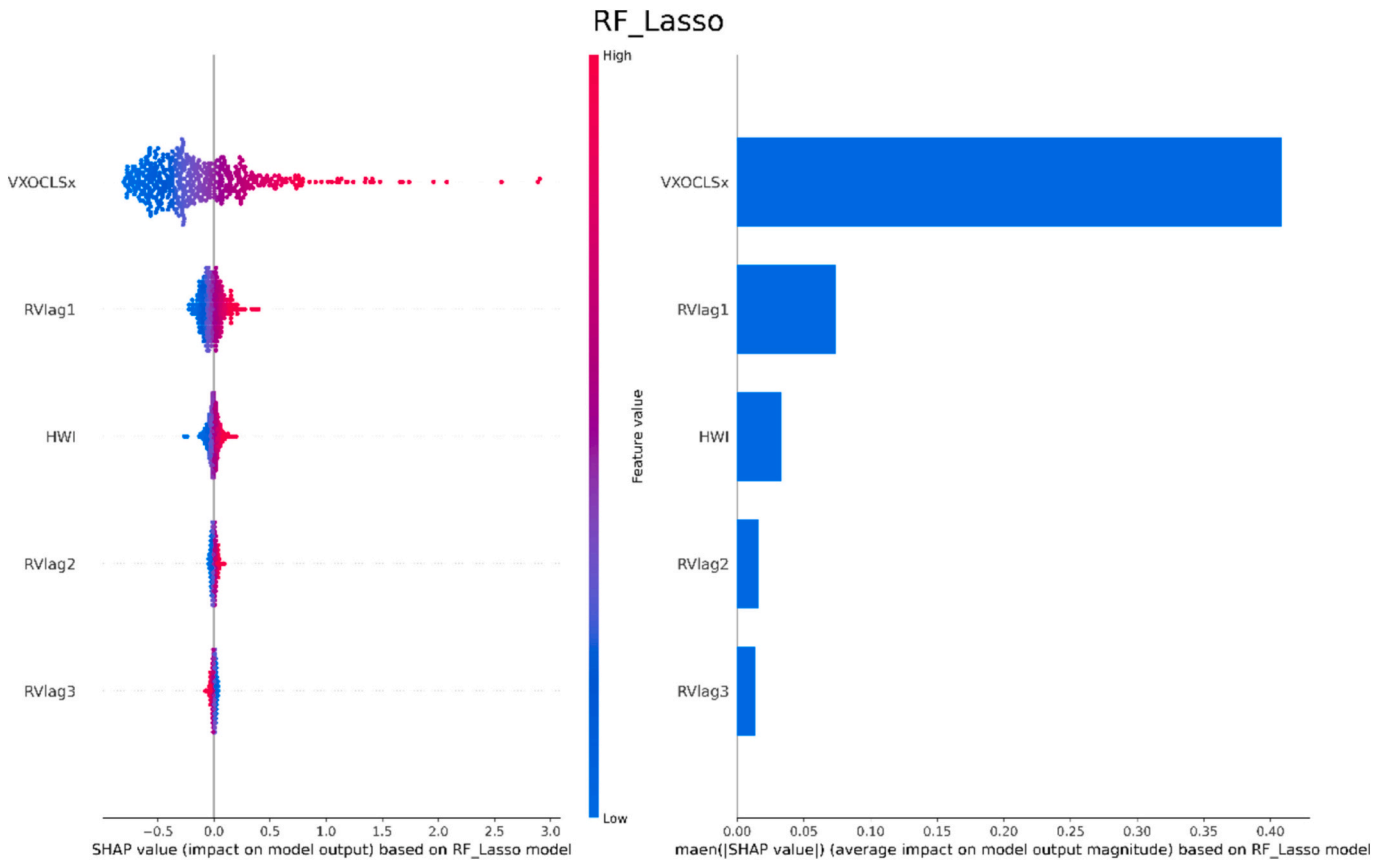

Fig. 6. This figure shows the model interpretability based on the SHAP method of RF-LASSO model.

Table 5
COVID-19 pandemic.

Forecasting models	R^2_{OOS}(%)	MSPE-Adj.	*p*-value
PCA	−20.3327	2.3400	0.0096
PLS	−4.2282	2.4580	0.0070
LASSO	**10.3096**	2.0299	0.0212
Ridge	**10.3096**	2.0299	0.0212
ENet	**10.3096**	2.0299	0.0212
ALASSO	**6.5529**	1.7354	0.0413
AENet	**6.5697**	1.7389	0.0410
GLASSO	**7.7013**	1.7728	0.0381
AGLASSO	**6.6051**	1.7425	0.0407
SGLASSO	**10.4959**	2.0128	0.0221
ASGLASSO	**6.5140**	1.7156	0.0431
SVR	**15.1514**	2.2940	0.0109
KNN	**19.7293**	2.4516	0.0071
ET	−14.4156	1.6956	0.0450
DT	−3.3511	1.1859	0.1178
RF	**26.3768**	1.7831	0.0373
Bagging	**28.1931**	1.8844	0.0298
Adaboost	**30.5455**	1.9504	0.0256
GBR	**21.6688**	2.5413	0.0055
XGB	**23.6724**	2.1243	0.0168
FCNN	**22.9293**	1.5067	0.0659
CNN	−9.4393	2.0833	0.0186
RNN	**17.3282**	2.1017	0.0178
LSTM	**11.8145**	2.2822	0.0112
C_LSTM	**37.3018**	2.2558	0.0120
RF-LASSO	**28.6291**	1.5607	0.0593

Notes: The table displays the post-sample assessment metrics for model performance during the COVID-19 pandemic, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

the ith sample as m_i, $m_{i,j}$ is the j-th feature and y_i represents the predictive value. The baseline is y_{base}. The SHAP value is:

$$y_i = y_{base} + f(m_{i,1}) + f(m_{i,2}) + \ldots + f(m_{i,k}) \tag{52}$$

where $f(m_{i,j})$ is the SHAP value of $m_{i,j}$. A positive SHAP value suggests that the feature contributes positively to the model's predictive power. Importantly, the SHAP approach not only measures the influence of each feature but also distinguishes between their positive and negative impacts. Figs. 5 and 6 show the results of the RF and RF-LASSO models, respectively. Our analysis reveals that the VXOCLSx variable emerges as the most influential predictor for gauging stock market volatility. Furthermore, it is observed that there is a positive correlation between VXOCLSx and the volatility of the stock market. Significantly, beyond the lagged values of RV, our findings also highlight the help-wanted index (HWI) for the United States as a crucial predictor for stock market volatility. This underscores the relevance of unemployment as an effective indicator for assessing labor market conditions and the dynamics of business cycles. Our conclusion is similar to the findings of (Christiano et al., 2021; Potestio, 2022).

5. Further discussion

5.1. COVID-19 pandemic

There is no doubt that the COVID-19 pandemic has already impacted costs and has had a destructive influence on the economy (Fernandez-Perez et al., 2021; McKibbin & Fernando, 2023; Niu et al., 2022). This study investigated the performance of the models during this pandemic. Table 5 shows the results of the models during the COVID-19 pandemic that started in January 2020. From Table 5, we find that 21 out of 26 models have significantly positive R^2_{OOS} values, showing that most of the

Table 6
Different business cycles.

Forecasting models	R^2_{OOS}(%)	MSPE-Adj.	*p*-value	R^2_{OOS}(%)	MSPE-Adj.	*p*-value
	Panel A: Expansions			Panel B: Recessions		
PCA	0.9851	1.6762	0.0469	−8.0954	2.8848	0.0020
PLS	0.1982	0.9485	0.1714	**2.1845**	2.9508	0.0016
LASSO	**8.0148**	3.9317	0.0000	**14.2541**	2.6514	0.0040
Ridge	**8.0148**	3.9317	0.0000	**14.2541**	2.6514	0.0040
ENet	**8.0148**	3.9317	0.0000	**14.2541**	2.6514	0.0040
ALASSO	**9.6550**	4.1885	0.0000	**11.7069**	2.4172	0.0078
AENet	**9.6403**	4.1874	0.0000	**11.7159**	2.4197	0.0078
GLASSO	**10.7442**	4.2967	0.0000	**13.9361**	2.5371	0.0056
AGLASSO	**10.2131**	4.3011	0.0000	**11.4228**	2.3890	0.0084
SGLASSO	**12.0153**	4.4777	0.0000	**15.8714**	2.7545	0.0029
ASGLASSO	**10.1499**	4.2923	0.0000	**11.3739**	2.3699	0.0089
SVR	**4.0708**	3.9177	0.0000	−37.4767	−0.0117	0.5047
KNN	−3.0386	2.7868	0.0027	−21.3312	0.7815	0.2172
ET	**6.6615**	4.9278	0.0000	**14.2671**	2.4544	0.0071
DT	**9.7800**	5.1675	0.0000	**9.7477**	2.1005	0.0178
RF	**27.5361**	5.9052	0.0000	**28.4007**	2.5225	0.0058
Bagging	**25.5737**	5.8364	0.0000	**27.8648**	2.5060	0.0061
Adaboost	**23.4532**	5.7553	0.0000	**30.7390**	2.3063	0.0105
GBR	−3.5651	2.5303	0.0057	−23.6148	0.6038	0.2730
XGB	**20.0286**	5.4630	0.0000	**20.1530**	2.2017	0.0138
FCNN	**19.8916**	5.3562	0.0000	**20.3300**	2.1731	0.0149
CNN	**6.8289**	5.4079	0.0000	**1.2577**	2.6429	0.0041
RNN	−0.5639	4.4570	0.0000	−7.1383	1.8712	0.0307
LSTM	−2.0379	4.3927	0.0000	−0.2428	2.6121	0.0045
C_LSTM	**14.1213**	5.1155	0.0000	**18.3106**	2.0006	0.0227
RF-LASSO	**31.5222**	6.7945	0.0000	**24.6515**	2.1948	0.0141

Notes: The table displays the post-sample assessment metrics for model performance during different business cycles, indicated by R^2_{OOS} and the MSPE-adjusted measure. The highest R^2_{OOS} value is highlighted with bold and underscore formatting.

models are still able to predict stock market volatility during this high volatility period, especially the ensemble learning, neural network, and new hybrid models.

5.2. Business cycles

Predictors can perform differently under different business cycles (Neely et al., 2014). This study examines how various macroeconomic variables predict stock market volatility based on business cycles. Table 6 presents the results of the study. First, during the expansion periods, we find that 22 out of 26 models have positive R^2_{OOS} values, showing that most machine learning models perform good. Second, the R^2_{OOS} value of the RF-LASSO model is 31.5222, which is the largest among the models and shows that the RF-LASSO model can best predict stock market volatility during expansion. During recession periods, the RF-LASSO model has a relatively weaker prediction performance than that during expansion periods. Similar results were also found during the recession periods.

6. Conclusion

From a comprehensive perspective, this study predicts stock market volatility based on over one hundred monthly macroeconomic variables, using a large set of machine learning models. We construct a new hybrid model integrating the random forest with the LASSO method. Several findings were obtained. First, most machine learning models, can predict stock market volatility. Second, we find that the random forest model is more efficient than commonly used machine learning models in terms of stock market volatility prediction. Third, the new hybrid model we constructed by combining the random forest and LASSO methods has the best predictive performance, showing that the combination of random forest and LASSO methods can further improve the forecasting accuracy of random forest. Fourth, robustness checks show that the new hybrid model can perform robustly in different circumstances.

Advancements in technology have led to an explosion in data availability, simultaneously giving rise to innovative business models that are revolutionizing investment approaches within the market. Looking ahead, machine learning is poised to offer extensive new opportunities, enhancing efficiency and enabling more precise risk management. In addition, studying the impact of macroeconomic variables on stock market volatility is of great significance to investors, policymakers, and market analysts. It provides an in-depth understanding of market trends and potential risks, helping investors make wiser investment decisions. At the same time, it provides a basis for policymakers when formulating macroeconomic policies that may affect the stock market, thereby promoting market stability and healthy economic development.

Data availability

Data will be made available on request.

Acknowledgments

The authors are grateful to the editor and anonymous referees for insightful comments that significantly improved the paper. This work is supported by the Natural Science Foundation of China [72071162; 72401042].

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.
Aven, T. (2016). Risk assessment and risk management: Review of recent advances on their foundation. *European Journal of Operational Research, 253*(1), 1–13.
Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications, 42*(20), 7046–7056.
Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning, 36*(1–2), 105–139.

Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies, 34*(2), 1046–1089.
Bollerslev, T., Hood, B., Huss, J., & Pedersen, L. H. (2018). Risk everywhere: Modeling and managing volatility. *The Review of Financial Studies, 31*(7), 2729–2773.
Bollerslev, T., & Mikkelsen, H. O. (1996). Modeling and pricing long memory in stock market volatility. *Journal of Econometrics, 1*(73), 151–184.
Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting, 38*(6), 600–619.
Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.
Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research, 70*, 245–317.
Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*.
Chen, W., Wang, Q., Hesthaven, J. S., & Zhang, C. (2021). Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of Computational Physics, 446*, Article 110666.
Chen, Y., Kumara, E. K., & Sivakumar, V. (2021). Investigation of finance industry on risk awareness model and digital economic growth. *Annals of Operations Research*, 1–22.
Chortareas, G., & Noikokyris, E. (2014). Monetary policy and stock returns under the MPC and inflation targeting. *International Review of Financial Analysis, 31*, 109–116.
Christensen, K., Siggaard, M., & Veliyev, B. (2023). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics, 21*(5), 1680–1727.
Christiano, L. J., Trabandt, M., & Walentin, K. (2021). Involuntary unemployment and the business cycle. *Review of Economic Dynamics, 39*, 26–54.
Clark, T. E., & West, K. D. (2007). Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics, 1*(138), 291–311.
Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. *Ensemble Machine Learning*, 157–175.
Fang, K., Wang, X., Zhang, S., Zhu, J., & Ma, S. (2015). Bi-level variable selection via adaptive sparse group lasso. *Journal of Statistical Computation and Simulation, 85*(13), 2750–2760.
Fang, T., Lee, T., & Su, Z. (2020). Predicting the long-term stock market volatility: A GARCH-MIDAS model with variable selection. *Journal of Empirical Finance, 58*, 36–49.
Fernandez-Perez, A., Gilbert, A., Indriawan, I., & Nguyen, N. H. (2021). COVID-19 pandemic and stock market response: A culture effect. *Journal of Behavioral and Experimental Finance, 29*, Article 100454.
Franses, P. H., & van Dijk, D. (1996). Forecasting stock market volatility using (nonlinear) GARCH models. *Journal of Forecasting*, 229–235.
Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Paper presented at the icml*.
Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232.
Gala, V. D., Pagliardi, G., & Zenios, S. A. (2023). Global political risk and international stock returns. *Journal of Empirical Finance, 72*, 78–102.
Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning, 36*, 3–42.
Girardin, E., & Joyeux, R. (2013). Macro fundamentals as a source of stock market volatility in China: A GARCH-MIDAS approach. *Economic Modelling, 34*, 59–68.
Goyal, A., & Welch, I. (2003). Predicting the equity premium with dividend ratios. *Management Science, 49*(5), 639–654.
Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223–2273.
Guiso, L., Sapienza, P., & Zingales, L. (2008). Trusting the stock market. *The Journal of Finance, 63*(6), 2557–2600.
Gupta, R., Nel, J., & Pierdzioch, C. (2023). Investor confidence and forecastability of US stock market realized volatility: Evidence from machine learning. *Journal of Behavioral Finance, 24*(1), 111–122.
Han, Y., Kim, J., & Enke, D. (2023). A machine learning trading system for the stock market based on N-period min-max labeling using XGBoost. *Expert Systems with Applications, 211*, Article 118581.
Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). Random forests. In *The elements of statistical learning: Data mining, inference, and prediction* (pp. 587–604).
HIHI, E. L.. (1996). Hierarchical recurrent neural networks for long-term dependencies. *Advances in Neural Information Processing System, 8*, 493–499.
Hung, M., & Yeh, A. J. (2023). Stock market alphas help predict macroeconomic innovations. *Macroeconomic Dynamics*, 1–35.
LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*(4), 541–551.
Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the Chinese stock market. *Journal of Financial Economics, 145*(2), 64–82.
Lewenstein, M., & Nowak, A. (1989). Fully connected neural networks with self-control of noise levels. *Physical Review Letters, 62*(2), 225–228.
Li, X., Li, G., & Yang, S. (2011). Adaptive elastic-net for general single-index regression models. In *Paper presented at the Nonlinear Mathematics for Uncertainty and its Applications*.
Liang, C., Li, Y., Ma, F., & Wei, Y. (2021). Global equity market volatilities forecasting: A comparison of leverage effects, jumps, and overnight information. *International Review of Financial Analysis, 75*, Article 101750.
Liang, C., Wang, L., & Duong, D. (2024). More attention and better volatility forecast accuracy: How does war attention affect stock volatility predictability? *Journal of Economic Behavior & Organization, 218*, 1–19.
Liu, Q., Tang, J., Li, D., & Xing, L. (2023). The role of bad-news coverage and media environments in crash risk around the world. *Journal of Empirical Finance, 72*, 488–509.
Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(1), 14–23.
Lu, X., Ma, F., Wang, J., & Zhu, B. (2021). Oil shocks and stock market volatility: New evidence. *Energy Economics, 103*, Article 105567.
Ma, F., Wang, J., Wahab, M., & Ma, Y. (2023). Stock market volatility predictability in a data-rich world: A new insight. *International Journal of Forecasting, 39*(4), 1804–1819.
Ma, F., Wei, Y., Liu, L., & Huang, D. (2018). Forecasting realized volatility of oil futures market: A new insight. *Journal of Forecasting, 37*(4), 419–436.
Ma, H., Gopalakrishnan, E. A., Menonab, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science, 132*, 1351–1362.
McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics, 34*(4), 574–589.
McKibbin, W., & Fernando, R. (2023). The global economic impacts of the COVID-19 pandemic. *Economic Modelling, 129*, Article 106551.
Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science, 21*(41), 89–93.
Nayak, A., Pai, M. M., & Pai, R. M. (2016). Prediction models for Indian stock market. *Procedia Computer Science, 89*, 441–449.
Neely, C. J., Rapach, D. E., Tu, J., & Zhou, G. (2014). Forecasting the equity risk premium: The role of technical indicators. *Management Science, 60*(7), 1772–1791.
Niu, Z., Demirer, R., Suleman, M. T., Zhang, H., & Zhu, X. (2024). Do industries predict stock market volatility? Evidence from machine learning models. *Journal of International Financial Markets, Institutions and Money, 90*, Article 101903.
Niu, Z., Ma, F., & Zhang, H. (2022). The role of uncertainty measures in volatility forecasting of the crude oil futures market before and during the COVID-19 pandemic. *Energy Economics, 112*, Article 106120.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 1*(42), 259–268.
Pesaran, M. H., Schleicher, C., & Zaffaroni, P. (2009). Model averaging in risk management with an application to futures markets. *Journal of Empirical Finance, 16*(2), 280–305.
Potestio, P. (2022). *Unemployment insurance over the business cycles modelling unemployment insurance: A survey (129-150)*. Springer.
Power, M. (2004). The risk management of everything. *The Journal of Risk Finance, 5*(3), 58–65.
Rasekhschaffe, K. C., & Jones, R. C. (2019). Machine learning for stock selection. *Financial Analysts Journal, 75*(3), 70–88.
Razmi, S. F., Bajgiran, B. R., Behname, M., Salari, T. E., & Razmi, S. M. J. (2020). The relationship of renewable energy consumption to stock market development and economic growth in Iran. *Renewable Energy, 145*, 2019–2024.
Rudin, C., & Wagstaff, K. L. (2014). *Machine learning for science and society* (Reprinted, (Reprinted,. *95* pp. 1–9). Springer.
Schmidhuber, J. (1992). Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation, 4*(1), 131–139.
Schwert, G. W. (1989). Why does stock market volatility change over time? *Journal of Finance*, 1115–1153.
Sharma, P. (2016). Forecasting stock market volatility using realized GARCH model: International evidence. *The Quarterly Review of Economics and Finance, 59*, 222–230.
Sheu, J. (2007). A coordinated reverse logistics system for regional management of multi-source hazardous wastes. *Computers & Operations Research, 34*(5), 1442–1462.
Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 231–245.
Tang, C., Garreau, D., & von Luxburg, U. (2018). When do random forests fail? *Advances in Neural Information Processing Systems, 31*.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
Vapnik, V. (1998). The support vector method of function estimation. *Nonlinear Modeling*, 55–85.
Wang, H., & Leng, C. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis, 52*(12), 5277–5286.
Watanabe, S. (2013). A widely applicable Bayesian information criterion. *The Journal of Machine Learning Research, 14*(1), 867–897.
Wu, D. (2016). Risk management and operations research: A review and introduction to the special volume. *Annals of Operations Research, 237*(1–2), 1–5.
Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 68*(1), 49–67.
Zhang, Y., Ma, F., & Wang, Y. (2019). Forecasting crude oil prices with a large set of predictors: Can LASSO select powerful predictors? *Journal of Empirical Finance, 54*, 97–117.
Zhang, Y., Wei, Y., Zhang, Y., & Jin, D. (2019). Forecasting oil price volatility: Forecast combination versus shrinkage method. *Energy Economics, 80*, 423–433.
Zou, H. (2006). The adaptive lasso and its Oracle properties. *Journal of the American Statistical Association, 101*, 1418–1429.
Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 67*(2), 301–320.
Zwikael, O., & Sadeh, A. (2006). Planning effort as an effective risk management tool. *Journal of Operations Management, 25*(4), 755–767.