RESEARCH



Forecasting VaR and ES by using deep quantile regression, GANs-based scenario generation, and heterogeneous market hypothesis



Jianzhou Wang¹, Shuai Wang^{1*}, Mengzheng Lv¹ and He Jiang²

*Correspondence: vvs09061513@163.com

 ¹ School of Statistics, Dongbei University of Finance and Economics, Dalian, China
 ² School of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China

Abstract

Value at risk (VaR) and expected shortfall (ES) have emerged as standard measures for detecting the market risk of financial assets and play essential roles in investment decisions, external regulations, and risk capital allocation. However, existing VaR estimation approaches fail to accurately reflect downside risks, and the ES estimation technique is quite limited owing to its challenging implementation. This causes financial institutions to overestimate or underestimate investment risk and finally leads to the inefficient allocation of financial resources. The main purpose of this study is to use machine learning to improve the accuracy of VaR estimation and provide an effective tool for ES estimation. Specifically, this study proposes a VaR estimator by combining quantile regression with "Mogrifier" recurrent neural networks to capture the "long memory" and "clustering" properties of financial assets; while for estimating ES, this study directly models the quantile of assets and employs generative adversarial networks to generate future tail risk scenarios. In addition to the typical properties of financial assets, the model design is also consistent with heterogeneous market theory. An empirical application to four major global stock indices shows that our model is superior to other existing models.

Keywords: Value at risk, Expected shortfall, Quantile regression, Recurrent neural networks, Generative adversarial networks

Introduction

In the ongoing credit and financial crises, it is essential to manage risk using appropriate measurement tools. Value at risk (VaR) is a widely used risk measure in financial institutions owing to its easy calculation and clear definition. However, the drawbacks of VaR (Artzner et al. 1999; Kwon 2021) are apparent: (1) VaR does not measure the left-tail risk beyond the quantile at the desired level and (2) VaR is not a consistent risk measure and does not satisfy some desirable properties such as subadditivity and convexity. These drawbacks cause investors and risk managers to overestimate or underestimate risk. In the recent Basel Accords, expected shortfall (ES) replaced VaR as the standard measure



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativecommons.org/licenses/by/4.0/.

of market risk, making it the most popular risk measure for financial institutions and investors (Acerbi and Tasche 2002). ES, in addition to having many other desirable properties, is a consistent risk measure defined as the conditional mean of the loss over VaR at a given confidence level (Rockafellar and Uryasev 2000). Although the ES employed in Basel III and IV can provide more information about the left tail of assets, estimation is inherently challenging, as ES is not elicitable, which means there exists no scoring function for which the expectation is minimized by the true ES (Gneiting 2011).

Classical approaches for forecasting VaR can be divided into three major categories: nonparametric, parametric, and semi-parametric. The nonparametric approach does not require assumptions regarding the distribution of returns. Historical simulation is the primary representative of this category, where the empirical distribution of historical returns is used to calculate VaR. Although the calculation complexity of this approach is relatively low, it cannot capture fluctuations that do not exist in the historical window used (Chang et al. 2003). To estimate VaR using a parametric approach, it is necessary to define an effective model of return distribution. Some well-known methods in this category are the variance-covariance model and many GARCH-type models. However, the distribution assumptions of this method (such as Gaussian and Student's t distributions) are not applicable to most financial time-series data. Semi-parametric approaches to VaR forecasting include those that use extreme value theory (Ener et al. 2012) and those that directly model the conditional quantile for a chosen probability level using quantile regression (QR), such as conditional autoregressive VaR (CAViaR) modeling (Engle and Manganelli 2004). In empirical studies on VaR forecast accuracy, the CAViaR models have performed well (Ener et al. 2012). Given the performance of QR in VaR estimation, scholars have combined it with machine learning methods, such as quantile regression neural networks (QRNN) and LASSO-QR, to further improve forecast accuracy. Moreover, there are inherently suitable methods for forecasting financial time series using deep learning such as recurrent neural networks (RNNs) and their variants. In particular, the MogLSTM proposed by Melis et al. (2019) has received widespread attention owing to the interaction operation between the hidden layer state and the current input first. We argue that through this interaction operation, the stored historical return information can be sufficiently combined with the currently available information to mimic long memory and nonlinear dependencies as well as the volatility clustering of returns. In addition, the forget gate in the MogLSTM cell can discard historical information that does not contribute to risk estimation, thereby avoiding serious clustered VaR violations. Therefore, we combined QR with a Mogrifier long short-term memory (MogLSTM) and Mogrifier gated recurrent unit (MogGRU) (Qin 2020) to propose two new deep learning QR (deep-QR) models: QRMogLSTM and QRMogGRU. For ease of expression, we call MogLSTM and MogGRU the Mogrifier recurrent neural networks (MogRNNs).

ES forecasts can be produced as byproducts of many VaR forecasting methods. Historical simulation and kernel density estimation in the nonparametric category can estimate ES by generating density forecasting. This is also the case for parametric approaches that involve a model for conditional variance, such as the GARCH model, and a distributional assumption. ES has always been regarded as not elicitable, whereas Fissler et al. (2015) show that VaR and ES are jointly elicitable. Therefore, another recently popular method is to jointly estimate VaR and ES based on the AL scoring function (Taylor 2019). Although machine learning methods can achieve accurate VaR estimations by combining QR, they provide no apparent means of producing ES forecasts. There is no scoring function for which the expectation is minimized by the true ES (Gneiting 2011) to construct supervised learning directly. In recent years, many data-driven methods based on Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) have been utilized to generate renewable resource scenarios, particularly for wind power (see, Ma et al. 2013; Liang and Tang 2020; Yuan et al. 2021). Although the potential benefits of the physical process from wind energy to wind power generation are obvious, one of the main obstacles to its implementation is the uncertainty of predicting meteorological variables. Therefore, researchers have developed data-driven methods for GANs that can generate scenarios based on historical data related to meteorological variables, and achieve satisfactory results for uncertainty prediction. In the field of risk management, a few studies have applied GANs to capture the uncertainty of financial asset returns and prices and obtain VaR estimations (see, Zhu et al. 2021; Fatouros et al. 2022). Although these generative models accurately portray variable uncertainty without the need to fit probabilistic models of stochastic variables, they do not provide an efficient way to estimate ES. Inspired by the above generative model-based research, this study employs GANs to specifically model uncertainty scenarios in the left tail of asset returns, and accordingly proposes an ES estimation method based on scenario generation. Specifically, after generating tail risk scenarios, ES can be estimated by calculating the arithmetic average of the risk scenarios below the VaR. This ES estimation method based on GANs risk scenario generation, which is interesting and promising, has never been explored.

In addition to the aforementioned VaR and ES modeling methods, the hypothesis of a heterogeneous market (Müller et al. 1993) potentially contributes to the estimation of risk measures. Different participants in a heterogeneous market have different time horizons and dealing frequencies, and can have different degrees of risk aversion, institutional constraints, and transaction costs. The risk measures in a heterogeneous market can be affected by participants with different dealing frequencies. In the VaR and ES estimation processes, we consider the heterogeneous market hypothesis as a guiding theory in the design of our estimation framework.

According to the pertinent literature, most existing VaR and ES methods face various challenges. First, traditional methods and simple machine learning models find it difficult to capture long memory and nonlinear dependencies in financial time series, which leads to the low accuracy of these models in the estimation of VaR and ES. The second challenge is severe VaR violations during which the asset realizes a loss exceeding the VaR value owing to dependencies between VaR forecasts, especially for the 99% confidence level. This is often the case when sharp market plunges occur, especially for developed markets (Žiković and Filer 2013). Third, although machine learning methods can achieve accurate VaR estimations by combining QR, they provide no apparent means of producing ES forecasts. Finally, financial market risk can be affected by participants with different time horizons and dealing frequencies, and existing risk-forecasting methods often ignore the importance of these factors. These challenges of VaR and ES methods reflect additional motivations for the framework proposed in this study.

In summary, this study introduces a data-driven framework that forecasts the VaR and ES of assets, addressing the aforementioned challenges with the following key innovations:

- (1) Two VaR estimation methods based on deep learning: QRMogLSTM and QRMog-GRU. These two methods not only mimic long memory and nonlinear dependencies to capture rare market events, but also discard unimportant historical information to avoid clustered VaR violations.
- (2) An ES estimation model based on GANs to solve the problem in which QR-based machine learning methods have difficulty generating ES estimates.
- (3) The use of the decomposition-aggregation mechanism to implement risk measure forecasting to capture the behavior of market participants with multiple time scales. Additionally, this study considers the heterogeneous market hypothesis as theoretical guidance in the design of the estimation model.

Related work

In finance, there is a growing interest in QR, with a particular focus on VaR models. There are two reasons for this: (1) QR provides a complete characterization of the random relationship between variables and (2) QR offers a more robust and thus more effective estimation in some non-Gaussian settings. The majority of the QR literature has focused on statistical models that generate future VaR in fixed model forms. For example, Koenker and Bassett (1978) propose a linear estimation process for conditional quantiles, extending the ordinary regression model by setting the loss function to quantile loss. Engle and Manganelli (2004) propose another quantile estimation method based on the linear QR technique to model the quantile directly, namely CAViaR. However, the return data of the real stock market are usually nonlinear and do not satisfy a normal distribution, and it is difficult to find suitable functional forms for the linear QR and CAViaR models (Huang 2013). To solve these estimation problems, Taylor (2000) applies the ORNN proposed by White (1992) to estimate conditional quantiles. The empirical results prove that the QRNN outperforms traditional GARCH-class models in terms of forecasting performance. The motivation for using QRNN to estimate VaR is clear: to find a suitable nonlinear functional form for the QR process with the powerful nonlinear mapping capability of artificial neural networks. Unfortunately, a QRNN is essentially a feedback-type neural network, and thus suffers from overfitting, underfitting, and a tendency to fall into local optimal points (Qiu and Song 2016). Inspired by Taylor, the following research idea is clear to scholars: extend the better-performing point forecasting model to the field of quantile estimation. For example, Takeuchi et al. (2006) and Li et al. (2007) proposed QR-SVM, and Xu et al. (2016) applied this method to VaR estimation and concluded that QR-SVM outperformed traditional GARCH-like and linear QR models. Nguyen et al. (2020) also propose the use of LASSO-QR to study tail risk in the cryptocurrency market.

With the development of deep learning, scholars have again turned their attention to deep neural networks, especially RNNs (Wang et al. 2022b) and their variants. For example, Wang et al. (2020) estimated quantiles using a QR long short-term memory

network (LSTM) and constructed forecasting intervals from the upper and lower quantile estimation results. However, few studies have focused on deep-QR to estimate the VaR. This may be because the financial sequence is more volatile, resulting in complex neural networks that are difficult to train and need a large number of hyperparameters to be set appropriately.

ES is a standard risk measure in the recent Basel Accord, while its related modeling work is less. Mainly because ES is not an elicitable measure, constructing a loss function for its estimation process is challenging (Cai and Wang 2008; Grabchak and Christou 2021). Scholars such as Du and Escanciano (2017) and Patton et al. (2019) have explored ES estimation approaches. Recently, a popular method has been proposed to jointly estimate VaR and ES based on the research theories of Fissler et al. (2015). For example, Meng and Taylor (2020) and Merlo et al. (2021) jointly estimated VaR and ES using CAV-iaR-type models based on the AL scoring function. In addition, the current literature on risk measurement modeling using generative models focuses only on VaR estimation without providing ideas for ES estimation. For example, Zhu et al. (2021) and Fatouros et al. (2022) employed GANs to generate price/return scenarios for future assets and obtained VaR by calculating the quantiles of all the scenarios. These methods are the same as scenario generation in the field of renewable energy (Li et al. 2020) and generally follow two steps: (1) obtaining the probability distribution of the sequence itself or the forecasting error, and (2) sampling the scenarios from the statistical distribution.

In a broad strand of the financial literature, the hypothesis of a heterogeneous market mainly guides the forecasting of asset volatility, such as the heterogeneous autoregressive (HAR) model (Corsi 2009). Decomposition-aggregation forecasting is a method that captures multi-timescale behaviors for the prediction of financial asset returns. The application scenarios of decomposition-aggregation forecasting in finance include, but are not limited to futures (Jiang et al. 2021; Guo et al. 2022), stock (Deng et al. 2022; Wang et al. 2022a), and cryptocurrency (Parvini et al. 2022) markets. The research results in the above studies show that forecasts based on decomposition-aggregation learning can improve accuracy. Moreover, the main idea of decomposition-aggregation forecasting is compatible with the heterogeneous market hypothesis.

Based on the above, the main contribution of the current study is fourfold: First, we propose a novel probabilistic framework for VaR estimation based on QRMogLSTM and QRMogGRU. This framework performs better in terms of both VaR 95% and VaR 99% than prevalent VaR estimation methods. Second, we developed an ES estimation approach based on GANs, providing future risk scenarios based on the generation technology of tail distribution, and solving the problem of machine learning methods having difficulties in producing ES forecasts. Various evaluation indices and statistical tests prove the validity of the ES forecasting approach. Third, we consider investors' different time horizons and dealing frequencies based on the heterogeneous market hypothesis. It captures the investment behaviors of short-, medium-, and long-term investors and does not significantly increase the computational burden. Finally, we explore a relatively complete QR model space using CAViaR-type models, QRNN, LASSO-QR, QR-SVM, QR-tree models, and QR-deep learning as benchmark models. The backtesting results of all the models were compared against four main stock indices, exploring questions that contribute to our understanding of the accurate estimation of risk measures.

VaR and ES estimation framework

In this study, we extend the decomposition-aggregation strategy, state-of-the-art deep learning methods, and Bayesian optimization technology to the field of risk measures estimation. These technologies are compatible with the properties of financial assets. The corresponding mathematical principles and financial theories are presented below.

The hypothesis of a heterogeneous market

Müller et al. (1993) proposed the hypothesis of a heterogeneous market, which states that different participants in a heterogeneous market have different time horizons and dealing frequencies. Therefore, the time horizon of participants has a "fractal" structure, which consists of short-, medium- and long-term components. Each component has its own reaction time to events or news, and different degrees of risk aversion, institutional constraints, and transaction costs. We argue that the risk measure of financial assets is likewise affected by the unique dealing frequency of heterogeneous participants. Therefore, we incorporate the heterogeneous market hypothesis into the design of our estimation framework. We use the decomposition-aggregation strategy ("Decomposition and aggregation based on investor heterogeneity" section) to distinguish participants with different time horizons and dealing frequencies and employ MogRNNs ("Quantile regression Mogrifier RNNs (QRMogRNNs)" section) to generate risk estimation specifically.

Decomposition and aggregation based on investor heterogeneity

Forecasting technology based on decomposition-aggregation learning has been applied in many research fields, such as finance, energy (Wang et al. 2022c; Neshat et al. 2022), and the environment (Kim et al. 2022; Wang et al. 2022d). Research findings in these fields show that forecasts based on decomposition-aggregation learning can improve accuracy. In risk measure estimation, we argue that this mechanism can identify and extract asset sequences at different frequencies (P H and Rishad 2020) for analysis and forecasts. This mechanism is compatible with the heterogeneous market hypothesis that short-term investors affect signals with higher frequencies, whereas the trading behaviors of medium- and long-term investors affect those with lower frequencies. Therefore, this study extends decomposition-aggregation learning to the field of risk measure estimation. Specifically, we propose a real-time decomposition-aggregation approach that uses the available information to learn decomposition and aggregation rules. Whenever new data are known, they are added to the available information set and then decomposed and aggregated according to the rules.

Variational mode decomposition

The decomposition method used in this study is the variational mode decomposition (VMD). VMD (Dragomiretskiy and Zosso 2014) is a powerful signal decomposition algorithm that decomposes a complex signal into several intrinsic mode functions (IMFs) with a specific center frequency and bandwidth that are completely nonrecursive. In the decomposition mode of the VMD, we can redefine the IMF as

$$u_k(t) = A_K(t)\cos(\varphi_k(t)),\tag{1}$$

where $A_K(t)$ and $\varphi_k(t)$ indicate the instantaneous amplitude and phase of $u_k(t)$, respectively; $w_k(t)$ is the instantaneous angular frequency. From $w_k(t) = d\varphi_k(t)/dt > 0$

(Dragomiretskiy and Zosso 2014), we know that $\varphi_k(t)$ must be differentiable at least once and $\varphi'_k(t) > 0$.

The VMD algorithm is realized by solving the following constrained variable problem:

$$\begin{cases} \underset{\{u_k, w_k\}}{\operatorname{argmin}} \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \exp(-jw_k t) \right\|_2^2, \\ s.t. \sum_{k=1}^{K} u_k(t) = v(t) \end{cases}$$
(2)

where $v_k(t)$ is the input signal, $j^2 = -1$, and $\delta(t)$ represents the Dirac distribution function. The Lagrange multiplier is used to transform Eq. (2) into an unconstrained optimization problem:

$$\underset{\{u_k,w_k\}}{\operatorname{argmin}} \alpha \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] \exp(-jw_k t) \right\|_2^2$$

$$+ \left\| v(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), v(t) - \sum_{k=1}^{K} u_k(t) \right\rangle,$$

$$(3)$$

where $\lambda(t)$ is the Lagrange multiplier used to enhance the constraint and $\alpha > 0$ is the quadratic penalty factor. $\left\| \nu(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2$ is a quadratic penalty term that accelerates the convergence rate and ensures minimum squared error.

According to the ADMM optimization method (Bertsekas and DimitriP 1982; Dragomiretskiy and Zosso 2014), the update modes of u_k and w_k can be expressed by the following equations:

$$\hat{u}_{k}^{n+1}(w) = \frac{\hat{x}(w) - \sum_{i < k} \hat{u}_{i}^{n}(w) + \hat{\lambda}(w) / 2}{1 + 2\alpha(w - w_{k})^{2}},$$
(4)

$$\hat{w}_{k}^{n+1} = \frac{\int_{0}^{\infty} w \left| \hat{u}_{k}^{n+1}(w) \right|^{2} dw}{\int_{0}^{\infty} \left| \hat{u}_{k}^{n+1}(w) \right|^{2} dw},\tag{5}$$

where $\hat{u}_k^{n+1}(w)$, $\hat{x}(w)$, and $\hat{\lambda}(w)$ are the Fourier transforms of the signals $u_k^{n+1}(t)$, x(t), and $\lambda(w)$, respectively. Moreover, the stop condition of VMD is expressed based on the tolerance of the convergence criterion:

$$\hat{u}_{k}^{n+1}(w) = \sum_{k} \frac{\left\| \hat{u}_{k}^{n+1}(w) - \hat{u}_{k}^{n}(w) \right\|_{2}^{2}}{\left\| \hat{u}_{k}^{n}(w) \right\|_{2}^{2}} < \varepsilon.$$
(6)

This study chooses the VMD algorithm instead of the empirical mode decomposition (EMD)-class (Huang et al. 1996) algorithm, because VMD can fix the number of generated IMFs, thus avoiding the inconsistency between in-sample and out-sample decomposition results.

IMFs aggregation based on fuzzy entropy

To balance forecasting accuracy and time consumption, we do not model each IMF, but the subseries aggregated by all the IMFs. However, there is no clear division between different frequencies of financial assets (IMFs), and the boundary is fuzzy. Thus, we employ the fuzzy entropy and approximation criterion (Fu et al. 2020) to aggregate all the IMFs. The approximation criterion is defined in Eqs. (7) and (8):

criterion =
$$\frac{\max{(FE_i|i=1,...,l)} - \min{(FE_i|i=1,...,l)}}{l/2}$$
, (7)

$$\Delta F E_{ij} = \left| F E_i - F E_j \right| < criterion, \tag{8}$$

where *l* denotes the number of decomposed IMFs and $\{FE_i | i = 1, ..., l\}$ represents the fuzzy entropy value corresponding to the *i*th IMF.

Real-time decomposition and aggregation mechanism

Let O_t be the financial asset data, and M_t be the available information set at time t. The real-time decomposition-aggregation mechanism can be implemented as follows:

- Decompose O₁, O₂,..., O_{t-1} at t-1, into k IMFs using VMD and information of M_{t-1}, such that the *j*th IMF can be denoted as (*IMF_j*|M_{t-1}).
- (2) Calculate the fuzzy entropy of all decomposed IMFs, then acquire the aggregation rule $(H|M_{t-1})$, and obtain p subseries $\overrightarrow{F_1}, \overrightarrow{F_2}, \ldots, \overrightarrow{F_p}, (p \le k)$ according to the approximate criterion.
- (3) Calculate the *j*th IMF, that is, $(IMF_{t,j}|M_t)$ when new information is available at *t* using the VMD and information from M_t .
- (4) Calculate the fuzzy entropy $FE_t(IMF_{t,j}|M_t)$ of each IMF at *t* and apply $(H|M_{t-1})$ to $FE_t(IMF_{t,j}|M_t)$ to obtain the aggregated subseries $F_{t,1}, F_{t,2}, \ldots, F_{t,p}, (p \le k)$ at *t*.

Figure 1 shows the decomposition and aggregation results of the SPX500 index for the training set. As shown in Fig. 1, the frequency of subseries (Sub) 1 is low; some researchers have argued that this sequence is a trend sequence (Zhang et al. 2008). The frequency of Sub 2 is relatively high, which is the medium- and long-term impact brought to the market by particular events or the behavior of medium- and long-term investors. Subs 3 and 4 are high-frequency sequences influenced by short-term shocks (such as monetary policy and the release of US macroeconomic data) or short-term investor behavior.



Fig. 1 Decomposition and aggregation results of SPX500

In the subsequent estimation, we only needed to model the subseries $F_{t,1}, F_{t,2}, \ldots, F_{t,p}$, which are aggregated in real time from IMFs and summed up the estimation results to obtain the estimated values of the financial asset data.

Quantile regression Mogrifier RNNs (QRMogRNNs)

Forecasting VaR at significance level α at a future time is equivalent to estimating the τ th quantile. We called Mogrifier LSTM (MogLSTM) and Mogrifier GRU (MogGRU) as MogRNNs collectively. QRMogRNNs are realized by designing the loss functions of MogRNNs as quantile loss, which is consistent with Taylor's study (2000) of QRNN.

Mogrifier LSTM and Mogrifier GRU

We found that the Mogrifier (Mog) structure proved to have better performance owing to an interesting model improvement, that is, performing regular interactive operations on the current input and the previous hidden state. The advantages of adopting MogRNNs to estimate the risk measures of financial assets are reflected in the properties of financial assets and the heterogeneous market hypothesis.

Properties of financial asset The interaction between the previous hidden layer which stored the historical return information and the current input enables the model to learn the "clustering" of financial asset volatility. By inputting the interaction results into the LSTM or GRU structure, the model retains or discards the historical information to learn the long-term dependencies of the financial sequence.

Heterogeneous market hypothesis The MogRNNs model architecture coincides well with the heterogeneous market hypothesis, as reflected in two ways. First, the RNNs choose to retain or discard historical information. Information that does not contribute to risk estimation is discarded (such as the impact of short-term speculation on the long-run market), whereas information that contributes to risk estimation (such as medium- and long-term investment behavior) is retained in the hidden layer of the RNNs. Second, the Mog structure can drive retained long-term investment behavior and short-term investment behavior performs interactive operations to simulate the interaction between short-term investors and medium- or long-term investors.

In addition to the abovementioned advantages, the forget gate in the LSTM cell can discard historical information that does not contribute to the estimation, thereby avoiding serious clustered VaR violations. Therefore, we combined QR with MogL-STM and MogGRU to propose two new deep-QR models to model the quantiles of financial assets, namely, QRMogLSTM and QRMogGR.

As LSTM and GRU have similar cell structures, we introduce the structure of MogLSTM to explain the model improvement. A schematic of MogLSTM is presented in Fig. 2. Its mathematical principle is as follows:

To demonstrate the modifications of MogLSTM, the cell structure of LSTM is first presented as follows:



Fig. 2 Schematic of MogLSTM with r = 5

$$\mathbf{F} = \sigma \left(\mathbf{W}^{Fx} x + \mathbf{W}^{Fh} \mathbf{h}_{prev} + \mathbf{b}^{F} \right),$$

$$\mathbf{I} = \sigma \left(\mathbf{W}^{Ix} x + \mathbf{W}^{Ih} \mathbf{h}_{prev} + \mathbf{b}^{I} \right),$$

$$\mathbf{J} = \tanh \left(\mathbf{W}^{Jx} x + \mathbf{W}^{Jh} \mathbf{h}_{prev} + \mathbf{b}^{J} \right),$$

$$\mathbf{O} = \sigma \left(\mathbf{W}^{Ox} x + \mathbf{W}^{Oh} \mathbf{h}_{prev} + \mathbf{b}^{O} \right),$$

$$\mathbf{C} = \mathbf{F} \odot \mathbf{C}_{prev} + \mathbf{I} \odot \mathbf{J},$$

$$\mathbf{h} = \mathbf{O} \odot \tanh (\mathbf{C}),$$
(9)

where $\sigma(\cdot)$ is the sigmoid function; **F**, **I**, **J**, and **O** are gates of the LSTM cell; they decide which information will be retained or discarded. The current input to the LSTM *x* must be related to the previous hidden state \mathbf{h}_{prev} . Thus, MogLSTM performs iterative interaction operations on **x** and \mathbf{h}_{prev} in advance to obtain the modulated inputs \mathbf{x}^{\uparrow} and $\mathbf{h}_{prev}^{\uparrow}$. These inputs, \mathbf{x}^{\uparrow} and $\mathbf{h}_{prev}^{\uparrow}$, can be defined as \mathbf{x}^{i} and \mathbf{h}_{prev}^{i} respectively, and expressed as follows:

$$\begin{cases} \mathbf{x}^{i} = 2\sigma \left(\mathbf{Q}^{i} \mathbf{h}_{prev}^{i-1} \right) \odot \mathbf{x}^{i-2}, & \text{for } \text{odd} i \in [1, 2, \dots, r] \\ \mathbf{h}_{prev}^{i} = 2\sigma \left(\mathbf{R}^{i} \mathbf{x}^{i-1} \right) \odot \mathbf{h}_{prev}^{i-2}, & \text{for } \text{even} i \in [1, 2, \dots, r] \end{cases}$$
(10)

with $\mathbf{x}^{-1} = \mathbf{x}$ and $\mathbf{h}_{prev}^0 = \mathbf{h}_{prev}$. The iterative round (also known as Mog step), $r \in \mathbb{N}$, is a hyperparameter; r = 0 recovers standard LSTM. \mathbf{Q}^i and \mathbf{R}^i are matrices resulting from random initialization To reduce the number of additional model parameters, the \mathbf{Q}^i and \mathbf{R}^i matrices are decomposed into the products of the low-rank matrices: $\mathbf{Q}^i = \mathbf{Q}_{left}^i \mathbf{Q}_{right}^i$ with $\mathbf{Q}^i \in \mathbb{R}^{m \times n}$, $\mathbf{Q}_{left}^i \in \mathbb{R}^{m \times k}$, $\mathbf{Q}_{right}^i \in \mathbb{R}^{k \times n}$, where $k < \min(m, n)$ is the rank.

MogGRU can be modified by the same mechanism, that is, the hidden state \mathbf{h}_{prev} and current input \mathbf{x} are operated interactively before they are inputted into the GRU cells.

QRMogRNNs

We assume that $f(\mathbf{X}_i, \mathbf{\theta})$ indicates the forecasting model and $\mathbf{\theta}$ represents all the parameters of this model. Next, we design the loss function in Eq. (11), where the parameter $\mathbf{\theta}(\tau)$ is estimated corresponding to τ , so we can estimate VaR using the model $f(\mathbf{X}_i, \hat{\theta}(\tau))$ based on QR theory:

$$QL = \sum_{i=1}^{N} \max\left\{\tau\left(f(\mathbf{X}_{i}, \boldsymbol{\theta}(\tau)) - y_{i}^{act}\right), (\tau - 1)\left(f(\mathbf{X}_{i}, \boldsymbol{\theta}(\tau)) - y_{i}^{act}\right)\right\}.$$
(11)

The condition quantile of y_i^{act} obtained by QRMogRNNs (i.e., QRMogGRU and QRMogLSTM) can be formulated based on the aforementioned forward propagation of MogLSTM and MogGRU as follows:

$$Q(\tau|X_i) = f\left(X_i, \hat{\theta}(\tau)\right) = \sigma\left(\mathbf{W}^{Ox}(\tau)x + \mathbf{W}^{Oh}(\tau)\mathbf{h}_{prev}(\tau) + \mathbf{b}^{O}(\tau)\right),$$
(12)

where $\mathbf{W}(\tau)$ and $\mathbf{b}(\tau)$ indicate the weight matrix and bias corresponding to the quantile τ , respectively; $\mathbf{h}_{prev}(\tau)$ represents the hidden state with respect to τ .

Bayesian hyperparameter optimization and model selection

The importance of using Bayesian theory and Bayesian formulas to realize model selection and hyperparameter optimization is reflected in two ways: First, the deep learning model has many hyperparameters that affect the accuracy of the model, especially for MogL-STM and MogGRU structures, which both have an extra parameter, *r*. Second, according to the "No free lunch" theorem, while model A outperforms model B on one specific task, there must be another specific task on which A performs worse than model B. Therefore, we regard the type of neural network layer as a hyperparameter that has two choices: QRMogLSTM and QRMogGRU. We coded them as 0 and 1, respectively, to realize the model selection. Table 1 presents the hyperparameters to be optimized and their value ranges, and the Bayesian optimization (BO) can be implemented as follows:

5 71			
Hyperparameter	Symbol	Value	Determination method
Number of features	N _f	[1,5]	Trial and error approach
Number of hidden units	N _h	[5,50]	Bias-variance tradeoff
Mog step	r	[0,10]	Preset
Number of layers	NL	[1,4]	Bias-variance tradeoff
Layer type	LT	[0,1]	Model selection principle
Epoch	epoch	[50,500]	Trial and error approach

Table 1 Value range of hyperparameter to be optimized

BO is based on Bayes' theorem, described as

$$P\left(\Lambda^{UN}|\Phi\right) = \frac{P\left(\Phi|\Lambda^{UN}\right)P\left(\Lambda^{UN}\right)}{P(\Phi)},\tag{13}$$

where Λ^{UN} indicates the unknown information, $P(\Lambda^{UN})$ refers to the prior distribution, $P(\Phi|\Lambda^{UN})$ indicates the probability, and $P(\Lambda^{UN}|\Phi)$ denotes the posterior distribution.

The basic steps of BO are given as.

- (1) Build a surrogate probability model of the objective function.
- (2) Find the ideal hyperparameters on the surrogate.
- (3) Apply these hyperparameters to the true objective function to assess them.
- (4) Update the surrogate model incorporating the new results.
- (5) Repeat steps (2-4) until max iterations are evaluated.

After considering the layer type as a hyperparameter and introducing the extra parameter *r*, we have four alternative models to select:

- (1) When r = 0 and LT = 0, the QRLSTM model is selected.
- (2) When r = 0 and LT = 1, the QRGRU model is selected.
- (3) When r > 0 and LT = 0, the QRMogLSTM model is selected, and the hidden layer state and the current input are operated interactively *r* times.
- (4) When r > 0 and LT = 1, the QRMogGRU model is selected.

Other parameters of our model are set by default: learning rate = 0.01, optimizer = "Adam" and dropout ratio = 0.2.

Estimate ES based on scenario generation

Although ES has replaced VaR as the standard risk measure in the recent Basel Accord, there are few ES estimation methods because they cannot construct a loss function. This study employs the generative model GANs to estimate ES. Inspired by the direct quantile modeling of the well-known CAViaR model proposed by Engle and Manganelli (2004), we employed GANs as a powerful tool to generate asset-tail scenarios. The principle is that the estimated quantile values are inputted into the GANs to generate many future risk scenarios at the tail of the assets, then the ES can be obtained by averaging scenarios lower than the estimated VaR value.

Least squares generative adversarial network

Least squares generative adversarial network (LSGAN) is a variant of the GANs proposed by Mao et al. (2017). This model maintains the core structure of the GAN, which consists of a generator G and discriminator D. The goal of G is to generate new data that D cannot determine as true or fake by learning the underlying distribution of the training-set data. D is a binary classifier that determines whether the data come from the

training set or G by classifying the input data as true or false. During the iterative process, G and D alternately update their weights to improve their discriminative or generative power and generate an adversarial model to reach the Nash equilibrium when D cannot discriminate between true and false.

The input to the generator is a latent variable Z, which can be a simple random distribution such as a Gaussian distribution. The goal of G is to map Z to G(Z) such that it approximates the true distribution as closely as possible. The inputs to D are the fake data generated by G and the training set $\overline{\Lambda}_1^{train}$. Because neural networks can fit arbitrary functions, generators and discriminators are generally designed as multilayer neural networks. This is achieved by designing the loss functions $L^{(D)}(\theta^{(D)}, \theta^{(G)})$ and $L^{(G)}(\theta^{(D)}, \theta^{(G)})$ for D and G. The main improvement in LSGAN is the modification of the loss functions of G and D to the form of a squared error. In LSGAN, the mathematical expressions for $L^{(D)}(\theta^{(D)}, \theta^{(G)})$ and $L^{(G)}(\theta^{(D)}, \theta^{(G)})$ are as follows:

$$L^{(D)}\left(\boldsymbol{\theta}^{(D)},\boldsymbol{\theta}^{(G)}\right) = \frac{1}{2} \mathbb{E}_{\overline{\Lambda}_{1}}^{\text{train}} \sum_{p_{data}} \left[\left(D\left(\overline{\overline{\Lambda}_{1}}^{\text{train}}\right) - \beta \right)^{2} \right] + \frac{1}{2} \mathbb{E}_{Z} \left[\left(D(G(Z)) - \alpha \right)^{2} \right],$$

$$(14)$$

$$L^{(G)}\left(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}\right) = \frac{1}{2} \mathbb{E}_{Z}\left[\left(D(G(Z)) - \gamma\right)^{2}\right],\tag{15}$$

where α , β , and γ are predefined parameters. Minimizing the objective function minimizes the *Pearson* χ^2 scatter when $\beta - \gamma = 1$ and $\beta - \alpha = 2$ are satisfied such that $\alpha = -1$, $\beta = 1$, and $\gamma = 0$. For **D**, a smaller $L^{(D)}(\theta^{(D)}, \theta^{(G)})$ indicates that it is more capable of distinguishing true from fake. For **G**, a smaller $L^{(G)}(\theta^{(D)}, \theta^{(G)})$ indicates that it is more capable of falsifying.

Training GANs is difficult in practice because of the instability of GANs learning. The discriminator of the original GAN uses a sigmoid cross-entropy loss function, which may lead to the problem of gradient vanishing during the learning process. The LSGAN uses the least square loss function (LSF) as its discriminator. The idea is simple and efficient: the LSF can move fake samples to the decision boundary because it punishes samples that lie on the correct side of the decision boundary for a long time. Based on this characteristic, the LSGAN is comparable to the Wasserstein generative adversarial network (WGAN) in generating samples that are closer to real data; however, the training process and convergence speed of the LSGAN are faster than those of the complex WGAN.

In addition, we considered a two-time-scale update rule training strategy (Heusel et al. 2017) to optimize the learning process of the LSGAN. Specifically, this study employs the "Adam" optimizer with different learning rates for the discriminator network D and the generator G. G uses the low-speed update rule with the learning rate set at 0.0001, and D uses the relatively fast update rule with the learning rate set at 0.0005.

Although the LSGAN ensures that the punishment for the outlier sample is greater, which solves the problem of unstable (insufficient) GANs training, the following

limitations remain. Excessive punishment of outliers by LSGAN may lead to a decrease in the "diversity" of sample generation, and the generator may have the problem of gradient vanishing when the discriminator is excellent enough. This study provides ideas for GANs to estimate ES and employs the LSGAN to implement the empirical study. In future research, we will explore other GANs variants with better performance to realize ES estimation.

Generating the future risk scenarios using LSGAN

Training GANs can be difficult and there are many possible setups. Therefore, we referred to the recent literature (Zhu et al. 2021; Fatouros et al. 2022) on modeling VaR using GANs to construct the structure and parameter settings of the LSGAN in this study. We implemented a conditional LSGAN in which the first step was a pretraining process. Generator G first estimates the in-sample VaR values to learn the future risk scenarios according to the critic scores given by discriminator D. The second step involves an extra training and forecasting process using the rolling window mechanism. Generator G takes b periods of the previous VaR values for additional training to generate the tail distribution of the following future f periods. The data are represented by M containing a total of W periods of VaR, split into two parts: the M_b containing VaR for the b previous periods, and the M_f that contains the VaR for the f following periods. Therefore, the generator generates simulations \hat{M}_f using the previous VaR M_b and latent vector Z. In practice, the latent vector represents the unknown future events affecting the market index. A schematic of the conditional LSGAN is shown in Fig. 3.

The generator is composed of a conditioning network and a simulator network. The conditioning network takes the historical risk trend M_b followed by a data normalization (Norm) process and 1D convolutional layers (Conv layers), where convolution is done over the time dimension. This output is then flattened and followed by a fully connected layer (Dense). The conditioning network then takes the latent input *Z* and concatenates it with the input of the previous layer. This is followed by a fully connected layer (Dense) before it is reshaped and followed by 1D convolutional layers (Conv layers) to shrink it



Fig. 3 Schematic of LSGAN

down to the desired output shape. The output of the generator \hat{M}_{f} is a vector with simulated tail risk for a market index. The discriminator takes a concatenation of real data M_{b} and fake data \hat{M}_{f} . This is processed by normalization (Norm) and several 1D convolutional layers (Conv layers), flattened, and finally there is a fully connected output layer (Dense). The discriminator gives a critic score, assigning a larger value if it believes the data comes from the true distribution, and a smaller value if it believes it comes from the distribution learned by the generator.

Forecasting ES based on risk scenarios and estimated VaR

As mentioned above, this study used risk scenarios to calculate the ES. Risk scenarios (i.e., quantile scenarios) sample approximate representations of the tail uncertainty relationship of asset indicators (e.g., log returns) on a future date. Suppose we obtain M_{in} observations of the in-sample VaR forecast results provided by any method: $\overline{VaR}(\tau) = \{VaR_1(\tau), VaR_2(\tau), \ldots, VaR_{M_{in}}(\tau)\}$. We directly model the estimated $\overline{VaR}(\tau)$ to generate future risk scenarios that follow a tail distribution. The ES value at the specified confidence level can then be calculated by averaging the scenarios lower than the corresponding VaR. This study forecasts ES based on rolling window exercises, and the model is retrained for every specific period. Assuming that the rolling window size is *b*, the model receives additional training for every *f* period according to the previous *b* periods estimated VaR values, and the epochs of pretraining and additional training are E_1 and E_2 .

The detailed steps for estimating ES using the LSGAN scenario generation approach are as follows:

- (1) Input the estimated in-sample VaR values $\overline{VaR}(\tau) = \{ VaR_1(\tau), VaR_2(\tau), \dots, VaR_{M_{in}}(\tau) \}$ into the LSGAN model for pre-training. The epoch of LSGAN is $E_1 = 1000$. The generator and discriminator game against each other, and the pretrained LSGAN $LSGAN_{trained} = LSGAN(\overline{VaR}(\tau))$ can generate future risk scenarios when they reach the Nash equilibrium.
- (2) Let the trained LSGAN generate *N* risk scenarios: $S_G = \{S_G^1, S_G^2, \dots, S_G^N\}$ = *LSGAN*_{trained}(*Z*), where *Z* is the latent vector. The generated risk scenarios contain the uncertain information at the tail of the asset.
- (3) Forecast the out-of-sample VaR value, VaR_{QRMogRNNs}, at a given confidence level (95% in this study) for time *t* from the QRMogRNNs and select scenarios with $\left(\widetilde{S_G}|\widetilde{S_G} < \text{VaR}_{QRMogRNNs}\right)$ lower than VaR_{QRMogRNNs}.
- (4) Average the selected scenarios to obtain the future f periods ES values: $ES = E\left(\widetilde{S_G}|\widetilde{S_G} < \operatorname{VaR}_{QRMogRNNs}\right).$
- (5) Train the LSGAN every *f* period according to the previous *b* periods VaR and set the epoch as $E_2 = 500$.
- (6) Repeat (2-5) steps to estimate all the out-of-sample ES values.

Since ES is not elicitable (Gneiting 2011), we chose to first estimate the tail risk of assets (i.e., VaR) and then model the tail risk scenarios based on the historical VaR values to estimate ES. Another potential modeling method is to use GANs directly to generate return scenarios and then calculate the arithmetic average of return scenarios

below the VaR value as the ES value. We also verified this method but found the following problems: if the asset returns are modeled directly, the available tail risk scenarios will be relatively few, especially during crisis periods. Furthermore, the tail scenarios of returns may be higher than the VaR values produced by the VaR estimator, which can lead to inaccurate or even ineffective risk forecast models. The method proposed in this study can be flexibly combined with any VaR estimator to jointly forecast ES without the abovementioned problems.

Evaluation methodology

VaR backtesting is typically based on coverage, which measures the percentage of times returns exceed the estimated VaR at a probability level of α (Emenogu et al. 2020). This study implements five commonly used VaR backtesting tests and one loss function: the unconditional coverage (*UC*) test (also known as Kupiec's POF test) (Kupiec 1995), independence (*IND*) test, conditional coverage (*CC*) test (Christoffersen 1998), dynamic quantile (*DQ*) test (Engle and Manganelli 2004), and Lopez's magnitude loss function (*M*-Loss) (Lopez 1999).

(1) LR_{UC}

$$LR_{UC} = -2\ln\left(\frac{\alpha^{N_1}(1-\alpha)^{N_0}}{\tilde{\Lambda}^{N_1}\left(1-\tilde{\Lambda}\right)^{N_0}}\right) \sim \chi_1^2,\tag{16}$$

where N_1 and N_0 are the number of times the VaR estimate is and is not exceeded, respectively, α indicates the desired level, and $\tilde{\Lambda}$ is the Hit Ratio (*HR*), which can be calculated using $N_1/(N_1 + N_0)$.

(2) LR_{IND}

$$LR_{IND} = -2\ln\left(\frac{\left(1 - \tilde{\Lambda}_2\right)^{(n_{00} + n_{10})} \tilde{\Lambda}_2^{(n_{01} + n_{11})}}{\left(1 - \tilde{\Lambda}_{01}\right)^{n_{00}} \tilde{\Lambda}_{01}^{n_{01}} \left(1 - \tilde{\Lambda}_{11}\right)^{n_{10}} \tilde{\Lambda}_{11}^{n_{11}}}\right) \sim \chi_1^2,$$
(17)

where n_{jk} is the number of *j* values followed by a *k* value in the sequence (with j, k = [0, 0], [0, 1] or [1, 1], and 0 indicating that the actual return does not exceed the VaR estimate; otherwise, 1), and $\tilde{\Lambda}s$ are defined as

$$\tilde{\Lambda}_{01} = \frac{n_{01}}{n_{00} + n_{01}}, \ \tilde{\Lambda}_{11} = \frac{n_{11}}{n_{10} + n_{11}} \ and \ \tilde{\Lambda}_2 = \frac{n_{01} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}}.$$
 (18)

(3) LR_{CC}

$$LR_{CC} = -2\ln\left(\frac{\alpha^{n_1}(1-\alpha)^{n_0}}{\left(1-\tilde{\Lambda}_{01}\right)^{n_{00}}\tilde{\Lambda}_{01}^{n_{01}}\left(1-\tilde{\Lambda}_{11}\right)^{n_{10}}\tilde{\Lambda}_{11}^{n_{11}}}\right) \sim \chi_2^2.$$
 (19)

(4) DQ test

Engle and Manganelli (2004) proposed the DQ test, and the statistical *Hit* considered in this test is very similar to $\tilde{\Lambda}$ with $Hit = \tilde{\Lambda} - \alpha$. Constructing a regression model of this variable with its lagged term as well as the estimated VaR series and other variables that should be considered, the regression form, considering the 4th order lagged term, is expressed as

$$Hit_{t} = \beta_{0} + \beta_{1}Hit_{t-1} + \beta_{2}Hit_{t-2} + \beta_{3}Hit_{t-3} + \beta_{4}Hit_{t-4} + \beta_{5}VaR_{t} + u_{t},$$
(20)

where u_t takes the value of $-\alpha$ with probability $1 - \alpha$, and $1 - \alpha$ with probability α .

The matrix representation of Eq. (1) is $\mathbf{Hit} = \beta \mathbf{Hit}_{lag} + u$. The null hypothesis is constructed as in $H_0: \mathbf{\beta} = 0$. According to the least squares method, $\hat{\beta} = \left(\mathbf{Hit}'_{lag}\mathbf{Hit}_{lag}\right)^{-1}\mathbf{Hit}'_{lag}\mathbf{Hit} \sim N\left(0, \alpha(1-\alpha)\left(\mathbf{Hit}'_{lag}\mathbf{Hit}_{lag}\right)^{-1}\right)$. In the case of the null hypothesis H_0 , the DQ test statistic can be expressed as

$$DQ = \frac{\beta \hat{H} i t'_{\text{lag}} \mathbf{H} i \mathbf{t}_{\text{lag}} \hat{\beta}}{\alpha (1 - \alpha)} \sim \chi_6^2.$$
(21)

The DQ test statistic asymptotically follows a chi-square distribution with six degrees of freedom.

(5) Lopez's magnitude loss function

The magnitude loss (*M-Loss*) function not only considers the number of losses but also the magnitude of extreme tail events; thus, the *M-Loss* function is more in line with economic significance. The *M-Loss* function can be defined as

$$M - Loss = I\left(Y_t < \hat{Y}_t^{\alpha}\right) \left(1 + \left(Y_t - \hat{Y}_t^{\alpha}\right)^2\right),\tag{22}$$

where Y_t is the observed return, \hat{Y}_t^{α} is the estimated VaR, and $I(\cdot)$ is an indicator function.

(6) Assessment of ES forecasts

To evaluate ES forecasts, this study implemented six ES backtesting tests: conditional (ES_C), unconditional (ES_{UC}) (Du and Escanciano 2017), minimally biased absolute (MBA), minimally biased relative (MBR) (Acerbi and Szekely 2014, 2017), N&Z (Nolde and Ziegel 2017), and M&F tests (McNeil and Frey 2000).

The hypothesis test form of ES_{UC} is $H_0: P_t^{[\alpha]} = F_t^{[\alpha]}, \forall t; H_1: ES_{\alpha,t}^F \ge ES_{\alpha,t}$, for all t and $ES_{\alpha,t}^F > ES_{\alpha,t}$ for some $t; VaR_{\alpha,t}^F = VaR_{\alpha,t}$, for all t, where $P_t^{[\alpha]}$ is the distribution of the forecasting model, $F_t^{[\alpha]}$ is a real (unknowable) distribution of assets, and $ES_{\alpha,t}^F$ and $VaR_{\alpha,t}^F$ indicate the risk measure value along a F distribution. The hypothesis test form

of ES_C is $H_0: P_t^{[\alpha]} = F_t^{[\alpha]}, \forall t; H_1: ES_{\alpha,t}^F \ge ES_{\alpha,t}$, for all t and $ES_{\alpha,t}^F > ES_{\alpha,t}$ for some t; $VaR_{\alpha,t}^F \ge VaR_{\alpha,t}$, for all t. The other four hypothesis test forms were adopted unaltered from the original literature.

Flow of the proposed model

The primary goal of this study was to devise a compound model for VaR and ES forecasts. QRMogLSTM and QRMogGRU were implemented on PyTorch, whereas LSGAN was built on TensorFlow. The sequence flow of the proposed model is presented in Algorithm 1 and the overall process is as follows:

Step 1: Divide the stock market index datasets into training, validation, and testing sets. The training and validation sets are used as in-sample data, whereas the testing set is used as out-of-sample data. A more detailed division of the data is presented in "Data" section.

Step 2: Using the subseries generation mechanism introduced in "The hypothesis of a heterogeneous market" section, learn the decomposition and aggregation rules on the training set and generate subseries according to the rules for every time data outside the training set are obtained.

Step 3: Estimate the quantiles of the subseries using the QRMogRNNs based on the Bayesian hyperparameter optimization and model selection in "Quantile regression Mogrifier RNNs (QRMogRNNs)" section. After that, sum up the subseries estimation values to obtain the final VaR estimation results for the stock market indices.

Step 4: Model the estimated quantiles directly using the LSGAN such that the generator and discriminator are trained by a zero-sum game, and the ES is estimated according to the rolling estimation mechanism.

Step 5: Backtest the VaR and ES estimation results, including the five VaR tests, risk scenario generation evaluation, six ES tests, and two joint scoring functions.

It should be noted that since ES can only be jointly elicitable with VaR, the ES estimation method in this study also relies on VaR estimation; thus, there is an inherent inconvenience in the training of the model. Other ES estimation studies have encountered the same challenge. For example, Taylor (2019) estimated VaR values based on the CAViaRtype model and then used the VaR values within a rolling window to estimate the parameters of the formula capable of deriving ES values. This method also requires obtaining the next-period VaR estimated value VaR_{t+1} and then using a rolling window to select the VaR values as the training data to estimate the parameters of the jointly elicitable formula to obtain the next-period ES estimated value. In this study, the rolling window method was used to train the LSGAN and forecast the VaR. The VaR_{t+1} is estimated first, and then the rolling window is employed to select the VaR sequence to be inputted into the LSGAN as the training data. The ES value in the next period is estimated based on VaR_{t+1} and the risk scenarios generated by the trained LSGAN. **Input**: R, historical returns; α , risk measure confidence probability; S, the number of risk scenarios Output: VaR, Out-of-sample VaR estimation; ES, Out-of-sample ES estimation **Real-time decomposition-aggregation mechanism** /* T = length(R) and T_1 is the length of training set **For** $t = T_1 : T$ $X_{txN} = VMD(R_1 \cdots R_t)$ /* R_t indicates returns available at time t, and N is the number of IMFs $X_{i \times P} = FuzzyEntropy(X_{i \times N})$ /* P is the number of aggregated sub-series End VaR forecasting process Dataset division For p = 1:P $\left[X_{p}^{train}, X_{p}^{validation}, X_{p}^{test}\right] = split(X_{t\times P})$ End Hyperparameter optimization and model selection Build a surrogate probability model of the objective function. $[Model_1, paras_1, pre_1_validation] = QRMogRNNs(X_n^{train}, X_n^{validation}, paras)$ /* paras indicate the range values of hyperparameters which are shown in Table 1 For i = 2: E /* E is the max iterations of Bayesian optimization $[Model_i, paras_i, pre_i validation] = QRMogRNNs(X_p^{train}, X_p^{validation}, model_{i-1}, paras_{i-1})$ /* Find the ideal hyperparameters on the surrogate. Update surrogate model incorporating the new results. End Model training $[Trained _Model, VaR_n^{\alpha}_train] = QRMogRNNs([X_n^{train}, X_n^{validation}], model_E, paras_E)$ /* Trained Model is the optimized QRMogRNNs with optimal architecture and parameters. VaR forecasting VaR_n^{α} pre = QRMogRNNs(X_n^{test} , Trained Model) /* estimated VaR values of each sub-series $VaR^{\alpha} = \sum VaR_{p} pre /*$ sum up the VaR values of P sub-series ES forecasting process Model pre-training $LSGAN_{pre-trained}^{a} = LSGAN\left(\sum_{n=1}^{p} VaR_{p}^{a} _ train, parameters\right) /* pre-training of LSGAN$ ES forecasting For t = T - Lt + 1: T /* the length of testing set is Lt /* The LSGAN is trained every f periods extra according to the most recent b periods data $LSGAN_{extra-trained}^{\alpha} = LSGAN_{pre-trained}^{\alpha} \left(\left| VaR_{t-b+1}^{\alpha}, \dots, VaR_{t}^{\alpha} \right| \right)$ $S_G^t = LSGAN_{extra-trained}^{\alpha}(Z) /* Z$ is the latent vector $ES_t^{\alpha} = \frac{1}{S} \sum \left(S_G^t \mid S_G^t < VaR_t^{\alpha} \right)$ End Return VaR^{α} , ES^{α}

Algorithm 1 VaR and ES forecasting framework.

Empirical study

Data

We took a sample of 1,683 weekly log-return from RESSET for the FTSE100, N225, SPX500, and DAX stock market indexes. The samples ranged from January 5, 1990, to April 1, 2022. In the VaR estimation, we used the first 1,262 data points as in-sample data to learn the decomposition-aggregation rules, train QRMogLSTM and QRMog-GRU, and divide the validation set for Bayesian hyperparameter optimization and model



Table 2 Summary statistics of the weekly log-return of the four indexes

Index	Mean	STD	Skewness	Kurtosis	J-B test	ADF test
FTSE100	0.067	2.332	- 1.006	13.821	8494.452	- 43.631
N225	- 0.020	3.048	- 0.653	8.878	2540.792	- 42.111
SPX500	0.152	2.326	- 0.852	10.583	4236.150	- 44.315
DAX	0.124	3.065	- 0.825	8.981	2699.387	- 41.727

selection. The final 421 data points were used as out-of-sample data for the backtesting procedure. In the ES estimation, we used the estimated 1,262 in-sample VaR values to pre-train LSGAN and the last 421 data points as out-of-sample data for the backtesting procedure. In this study, the pretraining epoch E_1 was 1000, the rolling window size was b = 100, and the model was extra-trained with additional $E_2 = 500$ epochs for every f = 10 period. Figure 4a–d show the division of the dataset and the purpose of each part schematically from four aspects. The experiment was implemented on a personal computer with an AMD Ryzen 5 5600H six-core processor, with Radeon Graphics 3.30 GHz, 16 GB RAM, and a single NVIDIA GeForce GTX 1650 GPU.

Table 2 presents the summary statistics, and the results confirm the prevalence of financial asset characteristics such as high kurtosis and fat tails. In addition, the log-return of all indices has negative skewness; the null hypothesis of the normal distribution of the Jarque–Bera test is rejected, and the null hypothesis of the existence of the unit root of the Augmented Dickey-Fuller test is also rejected. These results drove us to employ QR combined with deep learning to predict the quantiles and LSGAN to capture the tail uncertainty information of the assets.

Out-of-sample VaR forecasting

In this section, the proposed model is compared with key benchmarks. The statistics in "Evaluation methodology" section are used to backtest the VaR estimation results at two probability levels, namely, $\tau = 0.05$ and $\tau = 0.01$. This study includes 14 benchmarks: (1) Historical Simulation (Hist); (2) Normal distribution method (Normal); (3) CAViaR series models: CAViaR-SAV, CAViaR-AS, CAViaR-IGARCH, CAViaR-Adaptive; (4) QRNN; (5) LASSO-QR; (6) QR-random forest (QR-RF) and QR-gradient boosting decision trees (QR-GBDT); (7) QR-SVM; (8) QR-convolutional neural networks (QR-CNN); (9) QR-LSTM and QRGRU. The CAViaR-type models and QR-SVM were implemented in R, and the other models were implemented in Python.

Before the backtesting procedure, we analyzed the results of the hyperparameter determination and model selection, and the results are presented in Table 3.

(1) From the perspective of model selection, *r* is always greater than 0, which indicates that QRMogLSTM or QRMogGRU is superior to the naive QRLSTM or QRGRU; On the other hand, the optimal models are different on a specific subset of a specific dataset, which is consistent with the "No free lunch" theorem, which shows the importance of selecting models based on different datasets.

The data frequency in this study was weekly, between daily and monthly, and the data volume was relatively moderate. Therefore, the number of times the LSTM was selected was slightly higher than that of the GRU. Suppose that we model the data at a higher frequency (daily). In this case, the dataset would be larger, the LSTM with more parameters would be fully optimized, and the performance of the LSTM would be expected to improve. In contrast, because the GRU has fewer parameters to train, if we model lower-frequency (monthly) data, the learning process of the GRU converges more easily than that of the LSTM, and its performance may be better.

(2) There are many cases where the number of optimal model layers is three or four, and the number of hidden layer units is more than 10, which shows that QR combined with a complicated deep neural network can improve the prediction performance.

(3) From S1 to S4, the number of optimal feature inputs to the proposed model decreased. S1 is a low-frequency sequence and the proposed model requires additional lag features. As a high-frequency sequence, S4 always takes one as its optimal

~ 1															
FTSE	100			N22	N225			SPX500				DAX			
S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4	S 1	S 2	S 3	S 4
2	2	1	1	2	2	2	1	2	1	1	1	1	1	1	1
80	48	98	43	89	71	45	100	70	94	83	67	35	61	79	39
2	6	4	2	3	8	1	1	3	2	10	6	8	2	2	4
1	3	3	4	3	3	4	4	1	2	4	4	3	4	2	3
1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0
205	373	108	346	218	178	165	356	438	347	434	157	273	152	228	90
4	2	2	1	3	1	1	1	2	2	1	1	2	1	1	1
21	81	60	88	47	44	24	63	42	72	70	8	100	100	100	24
6	3	4	8	10	4	1	5	10	2	9	6	1	1	3	8
3	3	1	3	3	2	1	4	1	2	3	3	4	4	2	3
1	0	1	0	0	0	1	0	0	1	0	0	1	1	1	1
300	434	162	456	248	115	122	330	133	119	137	500	155	500	81	269
	FTSE S 1 2 80 2 1 1 205 4 21 6 3 1 300	FTSE100 S1 S2 2 2 80 48 2 6 1 3 1 0 205 373 4 2 21 81 6 3 3 3 1 0 300 434	FTSE IOO S1 S2 S3 2 2 1 80 48 98 2 6 4 1 3 3 1 0 1 205 373 108 4 2 2 21 81 60 6 3 4 3 3 1 1 0 1 300 434 162	FTSE IOU S1 S2 S3 S4 2 2 1 1 80 48 98 43 2 6 4 2 1 3 3 4 1 0 1 0 205 373 108 346 4 2 2 1 21 81 60 88 6 3 4 8 3 3 1 3 1 0 1 0 13 21 81 60 88 6 3 4 8 3 3 1 3 1 0 1 0 300 434 162 456	FTSE 100 N22: S1 S2 S3 S4 S1 2 2 1 1 2 80 48 98 43 89 2 6 4 2 3 1 3 3 4 3 1 0 1 0 0 205 373 108 346 218 4 2 2 1 3 21 81 60 88 47 6 3 4 8 10 3 1 3 3 1 3 1 0 1 0 0 0 3 1 3 3 1 3 3 1 0 1 0 0 0 0 300 434 162 456 248 162	FTSE 100 N225 S1 S2 S3 S4 S1 S2 2 2 1 1 2 2 80 48 98 43 89 71 2 6 4 2 3 8 1 3 3 4 3 3 1 0 1 0 0 0 205 373 108 346 218 178 4 2 2 1 3 1 1 21 81 60 88 47 44 6 3 4 8 10 4 3 3 1 3 3 2 1 0 1 0 0 0 0 3 1 3 3 2 1 3 3 2 1 0 1 0 0 0	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	FTSE 100 N225 SPX500 DAX 2 2 1 1 2 2 2 1 51 52 53 54 51 52 53 54 51 52 53 54 51 52 53 54 51 52 53 54 51 52 53 54 51 51 52 53 54 51 51 52 53 54 51 51 52 53 54 51 51 52 53 54 51 2 6 4 2 3 8 1 1 3 2 10 6 8 1 3 3 4 3 3 4 4 1 2 4 4 3 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </td <td>FTSE 100 N225 SPX500 DAX 2 2 1 1 2 2 2 1</td> <td>FTSE 100 N225 SPX500 DAX 2 2 1 1 2 2 2 1</td>	FTSE 100 N225 SPX500 DAX 2 2 1 1 2 2 2 1	FTSE 100 N225 SPX500 DAX 2 2 1 1 2 2 2 1

 Table 3
 Hyperparameter values determined by Bayesian optimization

In this paper, the model selection is realized by different values of LT and r. S refers to sub-series, which is aggregated by decomposed IMFs according to fuzzy entropy and approximation criterion.

feature, indicating that it is more susceptible to short-term impacts. Furthermore, when $\tau = 0.01$ is used, the optimal feature number is always between one and two, which means that more attention should be paid to the impact of short-term shocks when measuring risk with a higher confidence level.

The next section analyzes the performance of the proposed model and key benchmarks in the real market; the evaluation results of the out-of-sample VaR forecasts of the four tests are presented in Tables 4, 5, 6 and 7. The backtesting results for the

	HR	LR _{UC}	LR _{IND}	LR _{cc}	DQ	M-Loss
FTSE100 ($\tau = 0.01$)					
Hist	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	21.6577 (0.0014)	0.5430
Normal	0.0238	5.8313 (0.0157)	0.4879 (0.4849)	6.3192 (0.0424)	16.8946 (0.0097)	0.6574
CAViaR-SAV	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	17.3118 (0.0082)	0.4599
CAViaR-AS	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	21.4963 (0.0015)	0.3109
CAViaR- IGARCH	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	20.5383 (0.0022)	0.4081
CAViaR-Adap- tive	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	21.9149 (0.0013)	0.3567
QRNN	0.0190	2.7446 (0.0976)	0.3107 (0.5773)	3.0553 (0.2171)	12.5440 (0.0509)	0.2695
LASSO-QR	0.0285	9.7430 (0.0018)	0.9032 (0.3419)	10.6462 (0.0049)	63.6024 (0.0000)	0.6429
QR-RF	0.0190	2.7629 (0.0964)	0.3114 (0.5767)	3.0744 (0.2149)	7.0202 (0.3189)	0.1465
QR-GBDT	0.0237	5.8313 (0.0157)	0.4878 (0.4848)	6.3191 (0.0424)	20.1541 (0.0026)	0.2352
QR-SVM	0.0047	1.4438 (0.2295)	0.0191 (0.8899)	1.4630 (0.4811)	67.9498 (0.0000)	0.3003
QR-CNN	0.0142	0.6879 (0.4068)	0.1739 (0.6766)	0.8618 (0.6499)	16.2832 (0.0123)	0.4346
QR-LSTM	0.0095	0.0097 (0.9212)	0.0769 (0.7815)	0.0866 (0.9575)	25.6281 (0.0002)	0.3166
QR-GRU	0.0095	0.0097 (0.9212)	0.0769 (0.7815)	0.0866 (0.9575)	25.6967 (0.0002)	0.2739
Proposed model	0.0095	0.0098 (0.9212)	0.0769 (0.7815)	0.0867 (0.9576)	1.5349 (0.9571)	0.0752
FTSE100 ($\tau = 0.05$	5)					
Hist	0.0190	10.9777 (0.0009)	0.3107 (0.5773)	11.2884 (0.0035)	12.2992 (0.0556)	0.6499
Normal	0.0238	7.4618 (0.0063)	0.4879 (0.4849)	7.9496 (0.0188)	9.3648 (0.1541)	0.6789
CAViaR-SAV	0.0689	2.8823 (0.0896)	0.5078 (0.4761)	3.3901 (0.1836)	19.4318 (0.0035)	0.8580
CAViaR-AS	0.0523	0.0494 (0.8241)	0.0234 (0.8785)	0.0728 (0.9643)	5.6001 (0.4694)	0.7585
CAViaR- IGARCH	0.0618	1.1688 (0.2797)	0.1009 (0.7508)	1.2696 (0.5300)	11.9362 (0.0634)	0.7417
CAViaR-Adap- tive	0.0523	0.0494 (0.8241)	-0.0486 (1.0000)	0.0008 (0.9996)	11.6363 (0.0706)	0.7584
QRNN	0.0808	7.1932 (0.0073)	0.0788 (0.7790)	7.2720 (0.0264)	20.2884 (0.0025)	0.6577
LASSO-QR	0.0404	0.8555 (0.3550)	0.1361 (0.7122)	0.9916 (0.6091)	21.8211 (0.0013)	0.8343
QR-RF	0.0667	2.2692 (0.1319)	0.0099 (0.9203)	2.2792 (0.3199)	7.9304 (0.2432)	0.4509
QR-GBDT	0.0689	2.8823 (0.0896)	0.0000 (0.9986)	2.8823 (0.2367)	13.4368 (0.0366)	0.5641
QR-SVM	0.0190	10.9777 (0.0009)	0.3106 (0.5772)	11.2884 (0.0035)	10.9918 (0.0886)	0.6224
QR-CNN	0.0617	1.1687 (0.2796)	0.2983 (0.5848)	1.4671 (0.4801)	13.6996 (0.0331)	0.5077
QR-LSTM	0.0237	7.4617 (0.0063)	0.4962 (0.4811)	7.9580 (0.0187)	9.0742 (0.1694)	0.5667
QR-GRU	0.0380	1.3605 (0.2434)	1.2676 (0.2602)	2.6281 (0.2687)	10.1684 (0.1177)	0.6394
Proposed model	0.0475	0.0509 (0.8215)	-0.0335 (1.0000)	0.0174 (0.9914)	9.8586 (0.1307)	0.2592

Table 4 Out-of-sample VaR forecast evaluation of FTSE100

In brackets are the *p* values corresponding to the statistics. $HR = N_1/(N_1 + N_0)$, N_1 and N_0 are respectively the numbers of times that the VaR estimate is and is not exceeded. Bold values mean that the null hypothesis is not rejected

Table 5 Out-of-sample VaR forecast evaluation of N225

	HR	LR _{UC}	LR _{IND}	LR _{CC}	DQ	M-Loss
FTSE100 ($\tau = 0.01$)						
Hist	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	17.7829 (0.0068)	0.2771
Normal	0.0190	2.7446 (0.0976)	0.3107 (0.5773)	3.0553 (0.2171)	28.1403 (0.0001)	0.4588
CAViaR-SAV	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	17.2207 (0.0085)	0.2356
CAViaR-AS	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	24.1412 (0.0005)	0.0776
CAViaR-IGARCH	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	27.8503 (0.0001)	0.2287
CAViaR-Adap- tive	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	29.3186 (0.0001)	0.3251
QRNN	0.0238	5.8313 (0.0157)	0.4879 (0.4849)	6.3192 (0.0424)	22.9050 (0.0008)	0.2766
LASSO-QR	0.0309	11.9641 (0.0005)	0.6840 (0.4082)	12.6481 (0.0018)	65.5847 (0.0000)	0.5607
QR-RF	0.0214	4.1974 (0.0404)	0.4252 (0.5143)	4.6227 (0.0991)	20.5688 (0.0021)	0.0059
QR-GBDT	0.0190	2.7629 (0.0964)	0.3666 (0.5448)	3.1295 (0.2091)	21.2311 (0.0016)	0.0140
QR-SVM	0.0190	2.7629 (0.0964)	0.3114 (0.5767)	3.0744 (0.2149)	24.1986 (0.0004)	0.3314
QR-CNN	0.0095	0.0097 (0.9212)	0.0769 (0.7815)	0.0866 (0.9575)	26.6151 (0.0001)	0.1799
QR-LSTM	0.0166	1.5704 (0.2101)	0.2372 (0.6261)	1.8077 (0.4049)	24.4211 (0.0004)	0.1601
QR-GRU	0.0095	0.0097 (0.9212)	0.0769 (0.7815)	0.0866 (0.9575)	26.6151 (0.0002)	0.1164
Proposed Model	0.0071	0.3846 (0.5351)	0.0432 (0.8354)	0.4278 (0.8074)	7.6041 (0.2686)	0.0539
FTSE100 ($\tau = 0.05$)						
Hist	0.0475	0.0509 (0.8215)	0.0026 (0.9594)	0.0535 (0.9736)	9.2246 (0.1613)	0.7780
Normal	0.0499	0.0000 (1.0000)	0.0027 (0.9587)	0.0027 (0.9987)	14.8019 (0.0219)	0.8230
CAViaR-SAV	0.0475	0.0509 (0.8215)	2.0008 (0.1572)	2.0517 (0.3585)	6.3994 (0.3800)	0.7570
CAViaR-AS	0.0570	0.4321 (0.5110)	2.9109 (0.0880)	3.3430 (0.1880)	6.6924 (0.3502)	0.5730
CAViaR-IGARCH	0.0428	0.4731 (0.4916)	1.6125 (0.2041)	2.0856 (0.3525)	9.5196 (0.1464)	0.7122
CAViaR-Adap- tive	0.0475	0.0509 (0.8215)	2.0008 (0.1572)	2.0517 (0.3585)	12.3425 (0.0548)	0.9100
QRNN	0.0784	6.1956 (0.0128)	1.4514 (0.2283)	7.6470 (0.0219)	11.1709 (0.0832)	1.2096
LASSO-QR	0.0475	0.0509 (0.8215)	0.0026 (0.9594)	0.0535 (0.9736)	9.5989 (0.1426)	0.7427
QR-RF	0.0500	0.0001 (0.9910)	2.0569 (0.1515)	2.0570 (0.3575)	6.1982 (0.4013)	0.1075
QR-GBDT	0.0571	0.4473 (0.5035)	2.7198 (0.0991)	3.1672 (0.2052)	4.7453 (0.5768)	0.3773
QR-SVM	0.0523	0.0545 (0.8153)	-0.0482 (1.0000)	0.0062 (0.9968)	11.0896 (0.0856)	0.7758
QR-CNN	0.0522	0.0493 (0.8241)	2.4334 (0.1187)	2.4827 (0.2889)	7.3555 (0.2892)	0.6974
QR-LSTM	0.0617	1.1687 (0.2796)	3.4339 (0.0638)	4.6027 (0.1001)	6.8096 (0.3388)	0.7412
QR-GRU	0.0427	0.4730 (0.4915)	1.6124 (0.2041)	2.0855 (0.3524)	10.9791 (0.0890)	0.5282
Proposed model	0.0333	2.7691 (0.0961)	0.9657 (0.3258)	3.7348 (0.1545)	9.6196 (0.1416)	0.2862

In brackets are the *p* values corresponding to the statistics. $HR = N_1/(N_1 + N_0)$, N_1 and N_0 are respectively the numbers of times that the VaR estimate is and is not exceeded. Bold values mean that the null hypothesis is not rejected

FTSE100 index are presented in Table 4, the N225 index in Table 5, the SPX500 index in Table 6 and the DAX index in Table 7.

The first objective was to compare our model with competing benchmarks using five VaR evaluation measures. To facilitate a comparison of the backtesting results of the different models, we report the number of test rejections and average M-Loss values in Table 8.

Looking at the evaluation results for all four indices at both quantile levels, we find that the proposed model is successfully backtested at the 1% significance level and 5% confidence level, except for the SPX500 index at the $\tau = 0.01$ level. Our model obtained the minimum value of the average M-Loss, meaning that the magnitude of extreme events

Table 6 Out-of-sample VaR forecast evaluation of SPX500

	HR	LR _{UC}	LR _{IND}	LR _{cc}	DQ	M-Loss
FTSE100 ($\tau = 0.01$)						
Hist	0.0214	4.1742 (0.0411)	6.5479 (0.0105)	10.7221 (0.0047)	57.3800 (0.0001)	0.3337
Normal	0.0285	9.7430 (0.0018)	4.2422 (0.0394)	13.9852 (0.0009)	110.5046 (0.0000)	0.6047
CAViaR-SAV	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	14.0880 (0.0287)	0.1270
CAViaR-AS	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	3.6195 (0.7280)	0.1107
CAViaR-IGARCH	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	1.9139 (0.9274)	0.1211
CAViaR-Adap- tive	0.0166	1.5705 (0.2101)	2.7692 (0.0961)	4.3397 (0.1142)	34.3068 (0.0000)	0.1808
QRNN	0.0190	2.7446 (0.0976)	0.3107 (0.5773)	3.0553 (0.2171)	18.8983 (0.0043)	0.1487
LASSO-QR	0.0238	5.8313 (0.0157)	1.4700 (0.2254)	7.3013 (0.0260)	39.3872 (0.0000)	0.6639
QR-RF	0.0167	1.5840 (0.2081)	0.2378 (0.6257)	1.8218 (0.4021)	3.3463 (0.7642)	0.0055
QR-GBDT	0.0285	9.7809 (0.0017)	0.8998 (0.3428)	10.6808 (0.0047)	31.8562 (0.0000)	0.0106
QR-SVM	0.0095	0.0088 (0.9251)	5.1193 (0.0236)	5.1282 (0.0769)	57.6851 (0.0000)	0.3503
QR-CNN	0.0095	1.5704 (0.2101)	0.2372 (0.6261)	1.8077 (0.4049)	2.9629 (0.8134)	0.1223
QR-LSTM	0.0166	1.5704 (0.2101)	2.7692 (0.0960)	4.3396 (0.1141)	35.2707 (0.0000)	0.2261
QR-GRU	0.0095	0.0097 (0.9212)	0.0769 (0.7815)	0.0866 (0.9575)	23.6436 (0.0006)	0.1555
Proposed model	0.0166	1.5705 (0.2101)	0.2373 (0.6262)	1.8078 (0.4050)	13.8936 (0.0309)	0.2413
FTSE100 ($\tau = 0.05$)						
Hist	0.0475	0.0509 (0.8215)	0.9928 (0.3191)	1.0437 (0.5934)	25.9359 (0.0002)	0.8358
Normal	0.0499	0.0000 (1.0000)	0.7718 (0.3797)	0.7718 (0.6799)	23.7367 (0.0006)	0.9056
CAViaR-SAV	0.0475	0.0509 (0.8215)	0.9928 (0.3191)	1.0437 (0.5934)	4.1866 (0.6515)	0.5249
CAViaR-AS	0.0546	0.1947 (0.6590)	0.4212 (0.5164)	0.6159 (0.7349)	4.2419 (0.6440)	0.4051
CAViaR-IGARCH	0.0499	0.0000 (1.0000)	0.7718 (0.3797)	0.7718 (0.6799)	11.5924 (0.0717)	0.4300
CAViaR-Adap- tive	0.0570	0.4321 (0.5110)	0.2883 (0.5913)	0.7205 (0.6975)	16.5270 (0.0112)	0.6931
QRNN	0.0594	0.7579 (0.3840)	3.1667 (0.0752)	3.9246 (0.1405)	18.1966 (0.0058)	0.5948
LASSO-QR	0.0451	0.2068 (0.6493)	1.2470 (0.2641)	1.4539 (0.4834)	21.3708 (0.0016)	0.8884
QR-RF	0.0619	1.1941 (0.2744)	3.4427 (0.0635)	4.6368 (0.0984)	4.9043 (0.5561)	0.0732
QR-GBDT	0.0642	1.6921 (0.1933)	0.4165 (0.5186)	2.1086 (0.3484)	6.9544 (0.3250)	0.0397
QR-SVM	0.0452	0.1969 (0.6571)	1.2410 (0.2652)	1.4380 (0.4872)	20.8298 (0.0019)	0.7870
QR-CNN	0.0403	0.8554 (0.3550)	1.8654 (0.1719)	2.7209 (0.2565)	14.6888 (0.0228)	0.5434
QR-LSTM	0.0427	0.4730 (0.4915)	1.5369 (0.2150)	2.0100 (0.3660)	13.7159 (0.0329)	0.7124
QR-GRU	0.0427	0.4730 (0.4915)	1.5369 (0.2150)	2.0100 (0.3660)	13.2865 (0.0387)	0.6553
Proposed model	0.0475	0.0509 (0.8215)	3.2571 (0.0711)	3.3080 (0.1913)	11.7756 (0.0672)	0.4115

In brackets are the *p* values corresponding to the statistics. $HR = N_1/(N_1 + N_0)$, N_1 and N_0 are respectively the numbers of times that the VaR estimate is and is not exceeded. Bold values mean that the null hypothesis is not rejected

beyond the forecast was small. Other benchmarks have significantly fewer successful backtesting scenarios, mainly because they cannot pass the DQ test at $\tau = 0.01$, and there are cases of risk overestimation or underestimation. The advantage of our model is also reflected in the VaR backtesting results at $\tau = 0.01$, which can be used to assess risk more accurately. Therefore, our model does not significantly underestimate extreme losses. In addition, it is worth noting that the CAViaR-AS model also outperforms other CAViaR models in terms of forecast performance, which is consistent with the findings of Merlo et al. (2021).

The above analysis, based on typical tests, shows that the proposed model can forecast risk accurately in more scenarios than the other 14 benchmarks. We also found

Table 7 Out-o	f-sample Va	R forecast e	valuation c	of DAX
---------------	-------------	--------------	-------------	--------

	HR	LR _{UC}	LR _{IND}	LR _{CC}	DQ	M-Loss
FTSE100 ($\tau = 0.01$)						
Hist	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	17.2559 (0.0084)	0.6477
Normal	0.0166	1.5705 (0.2101)	0.2373 (0.6261)	1.8078 (0.4050)	15.7729 (0.0150)	0.8753
CAViaR-SAV	0.0166	1.5705 (0.2101)	0.2373 (0.6261)	1.8078 (0.4050)	14.5064 (0.0245)	0.4353
CAViaR-AS	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	20.5229 (0.0022)	0.3184
CAViaR-IGARCH	0.0119	0.1451 (0.7033)	0.1205 (0.7285)	0.2656 (0.8757)	19.5541 (0.0033)	0.3134
CAViaR-Adaptive	0.0190	2.7446 (0.0976)	0.3107 (0.5773)	3.0553 (0.2171)	19.3534 (0.0036)	0.2989
QRNN	0.0071	0.3846 (0.5351)	0.0432 (0.8354)	0.4278 (0.8074)	1.2566 (0.9740)	0.1013
LASSO-QR	0.0143	0.6879 (0.4069)	0.1739 (0.6767)	0.8618 (0.6499)	26.1722 (0.0002)	0.7968
QR-RF	0.0214	4.1974 (0.0404)	0.3951 (0.5296)	4.5925 (0.1006)	24.2296 (0.0005)	0.6218
QR-GBDT	0.0214	4.1742 (0.0411)	0.3942 (0.5301)	4.5683 (0.1019)	20.0751 (0.0027)	0.6056
QR-SVM	0.0142	0.6879 (0.4068)	0.1739 (0.6766)	0.8618 (0.6499)	16.4757 (0.0114)	0.5806
QR-CNN	0.0166	1.5704 (0.2101)	0.2372 (0.6261)	1.8077 (0.4049)	14.8120 (0.0217)	0.4762
QR-LSTM	0.0142	0.6879 (0.4068)	0.1739 (0.6766)	0.8618 (0.6499)	16.3423 (0.0120)	0.4536
QR-GRU	0.0190	2.7445 (0.0975)	0.3106 (0.5772)	3.0552 (0.2170)	14.2153 (0.0273)	0.5161
Proposed model	0.0071	0.3846 (0.5351)	0.0432 (0.8354)	0.4278 (0.8074)	1.7891 (0.9380)	0.1812
FTSE100 ($\tau = 0.05$)						
Hist	0.0546	0.1947 (0.6590)	0.0639 (0.8004)	0.2587 (0.8787)	3.3210 (0.7676)	1.2893
Normal	0.0546	0.1947 (0.6590)	0.0639 (0.8004)	0.2587 (0.8787)	4.2481 (0.6431)	1.2728
CAViaR-SAV	0.0475	0.0509 (0.8215)	0.0026 (0.9594)	0.0535 (0.9736)	3.4985 (0.7442)	1.0248
CAViaR-AS	0.0499	0.0000 (1.0000)	0.0027 (0.9587)	0.0027 (0.9987)	0.5658 (0.9969)	0.9339
CAViaR-IGARCH	0.0546	0.1947 (0.6590)	0.4212 (0.5164)	0.6159 (0.7349)	4.4775 (0.6124)	0.9515
CAViaR-Adaptive	0.0475	0.0509 (0.8215)	0.0026 (0.9594)	0.0535 (0.9736)	8.2815 (0.2182)	1.2434
QRNN	0.0736	4.3996 (0.0360)	1.0393 (0.3080)	5.4388 (0.0659)	17.8002 (0.0068)	0.7259
LASSO-QR	0.0285	4.7707 (0.0290)	0.7060 (0.4008)	5.4767 (0.0647)	6.3422 (0.3860)	1.0813
QR-RF	0.0593	0.7579 (0.3839)	0.2019 (0.6531)	0.9598 (0.6188)	3.9701 (0.6807)	0.9914
QR-GBDT	0.0475	0.0509 (0.8215)	2.0008 (0.1572)	2.0517 (0.3585)	3.1034 (0.7958)	0.8558
QR-SVM	0.0427	0.4730 (0.4915)	0.0681 (0.7941)	0.5411 (0.7629)	1.1158 (0.9808)	1.1616
QR-CNN	0.0570	0.4321 (0.5109)	0.2883 (0.5912)	0.7204 (0.6975)	3.2423 (0.7778)	0.8453
QR-LSTM	0.0783	6.1955 (0.0128)	0.0010 (0.9744)	6.1966 (0.0451)	12.6532 (0.0488)	1.1551
QR-GRU	0.0285	4.7707 (0.0289)	0.7059 (0.4007)	5.4767 (0.0646)	5.9811 (0.4253)	0.7904
Proposed model	0.0333	2.7691 (0.0961)	0.9657 (0.3258)	3.7348 (0.1545)	4.9745 (0.5471)	0.3911

In brackets are the *p* values corresponding to the statistics. $HR = N_1/(N_1 + N_0)$, N_1 and N_0 are respectively the numbers of times that the VaR estimate is and is not exceeded. Bold values mean that the null hypothesis is not rejected

evidence of this in the sequence diagrams. To obtain more explicit images of how different models differ in their risk forecasts, we plot the series for only the three models that performed better in the above tests: CAViaR-AS, QRNN, and our proposed model. The circles in the plot represent the actual returns, with yellow and green representing positive and negative returns, respectively; the red circles represent the actual returns, where the losses exceed the forecasts of the proposed model. The analysis in Fig. 5 reveals that the risk forecast curve of CAViaR-AS is smoother, whereas those of the proposed model and QRNN are volatile. Moreover, the difference in performance between the models is not significant during periods of stable stock prices, whereas it is more significant in the aftermath of major financial crises such as the Chinese stock market crash, Brexit, and the breakout of COVID-19. In March 2020 (enlarged area in the figure), the global stock market showed violent fluctuations

Model	$\tau = 0.01$						τ = 0.05				
	UC	IND	СС	DQ	Mean (M-Loss)	UC	IND	СС	DQ	Mean (M-Loss)	
Hist	1	1	1	4	0.4504	1	0	1	1	0.8882	
Normal	2	1	1	4	0.6490	1	0	1	2	0.9201	
CAViaR-SAV	0	0	0	4	0.3144	0	0	0	1	0.7912	
CAViaR-AS	0	0	0	3	0.2044	0	0	0	0	0.6677	
CAViaR-IGARCH	0	0	0	3	0.2678	0	0	0	1	0.7088	
CAViaR-Adaptive	0	0	0	4	0.2904	0	0	0	1	0.9012	
QRNN	1	0	1	2	0.1990	2	0	2	3	0.7970	
LASSO-QR	3	0	3	4	0.6661	1	0	0	2	0.8867	
QR-RF	2	0	0	2	0.1949	0	0	0	0	0.4057	
QR-GBDT	3	0	2	4	0.2164	0	0	0	1	0.4592	
QR-SVM	0	1	0	4	0.3907	1	0	1	1	0.8367	
QR-CNN	0	0	0	3	0.3032	0	0	0	2	0.6485	
QR-LSTM	0	0	0	4	0.2891	2	0	2	2	0.7939	
QR-GRU	0	0	0	4	0.2655	1	0	0	1	0.6533	
Proposed Model	0	0	0	1	0.1379	0	0	0	0	0.3370	

Table 8 Summary of the backtesting results for each probability level for the stock indices. Number of test rejections for the four indexes

Smaller values are preferred



Fig. 5 Out-of-sample forecasts of VaR for the four stock market indexes. The top panels refer to $\tau = 0.01$ and the case of $\tau = 0.05$ is presented in the lower panels. Gray bands correspond to Chinese stock market crash (2015, 07–2016, 09), 2018 recession (2018, 01–2018, 06), Brexit (2018, 08–2019, 03), and COVID-19 outbreak (2020, 02–2020, 03)

owing to the COVID-19 pandemic, and stock prices fell sharply. The excess red circles also appear more intensively during this period, which indicates that the forecasting model does not accurately forecast sudden risks. Nevertheless, the difference between the actual and forecasted values of the proposed model was not significant, whereas the other models were more likely to overestimate the risk at $\tau = 0.01$ and underestimate it at $\tau = 0.05$, suggesting that our model is still significantly more accurate.

In addition to estimation accuracy, we also need to consider the time required to generate a VaR forecasting result, as our forecasting framework includes time-consuming tasks such as hyperparameter optimization, model selection, and deep learning training. Among these, BO is the most time-consuming process. We performed

τ	UC		сс	СС		DQ		Mean (M-Loss)	
	0.01	0.05	0.01	0.05	0.01	0.05	0.01	0.05	
Proposed Model	0	0	0	0	1	0	0.1379	0.3370	
-(DA)	0	0	0	0	2	1	0.2629	0.5036	
-(BO)	0	0	0	0	2	1	0.1862	0.4220	
-(Mog)	0	0	0	0	3	1	0.1143	0.4774	

Table 9 Summary of the backtesting results for ablation studies

-(DA), -(BO) and -(Mog) denote the removal of DA, BO and Mog from the proposed framework, respectively

a BO process with 50 iterations for each of the four subseries, each subseries took 196 s, and the time consumption for calculating the four components was 784 s. Nonetheless, BO determines the structure of our VaR estimation model and the corresponding parameter values over a period; therefore, we do not need to perform BO for every forecast. After the BO, the training time of our model was relatively short. It only took 1.84 s to obtain the VaR value of a subseries; therefore, a total of 7.36 s were required to obtain a forecasted VaR value. This time consumption is relatively short; therefore, our model can also be used for daily VaR estimations.

Finally, ablation studies were performed to evaluate the effects of the decomposition and aggregation technique (DA), BO, and Mogrifier structure (Mog) on the proposed model. We report the number of test rejections and average M-Loss values for the four indices. The results are summarized in Table 9. It can be observed that DA, BO, and Mog significantly improve the proposed model. In particular, DA preprocessing techniques and the Mog structure are more important than the BO. The proposed model, without hyperparameter optimization and model selection, can also achieve better riskforecasting results than the CAViaR-type models and QRNN.

Out-of-sample ES forecasting

This section evaluates the out-of-sample ES forecasting results. The risk scenarios generated by the LSGAN are assessed in "Evaluation of risk scenario generation results" section. Furthermore, the out-of-sample ES forecasting results are evaluated using statistical tests and scoring functions, as presented in "Backtesting ES with statistical tests and joint scoring functions" section.

Evaluation of risk scenario generation results

After pretraining, the LSGAN performs additional training based on the latest 100 VaR historical data at the 95% probability level and generates 500 quantile scenarios for outof-sample ES estimation. The model architecture and parameter settings of the LSGAN designed in this study are listed in Table 10.

The evaluation of scenario generation techniques commonly encompasses three primary categories (Li et al. 2020): (1) output-based evaluation, which leverages error metrics such as the mean squared error to gauge performance; (2) distribution-based evaluation, entailing assessment via computations of the energy score; and (3) eventbased evaluation, encompassing metrics such as coverage rate and correlation coefficients. We carefully selected the coverage rate (CR) and correlation coefficients (CC) of

Description	Value
Generator G	
Conditioning network	
Size of latent vector Z	2*1
Rolling window <i>b</i>	100
Forward window f	10
1D Conv layers	4
Input and output channels in all Conv layers	2*1
Dense layer output size	1
Simulator network	
Dense layer output size	<i>f</i> *1
Transpose Conv layers	2
Input channels in the first transpose Conv layer	4*1
Output channels in the first transpose Conv layer	2*1
Input channels in the second transpose Conv layer	2*1
Output channels in the second transpose Conv layer	1
Conditioning and Simulator networks	
Layers' activation function	Relu
Conv and transpose conv kernel length	5
Conv and transpose conv stride	2
Discriminator D	
Convolution layers	4
Input channels for the <i>k</i> th Conv layer	2^{k-1}
Output channels for the <i>k</i> th Conv layer	2 ^{<i>k</i>}
Layers' activation function	Leaky_Relu
Conv and transpose conv kernel length	5
Conv and transpose conv stride	2

Table 10 List of LSGAN parameter

the event-based evaluation method for the initial evaluation of the scenario generation results. This is because using other evaluation methods requires obtaining actual observations, which do not exist for risk measures (i.e., VaR and ES). To calculate the ES value at a certain significance level based on the risk scenarios, it is necessary to ensure that the ES value is lower than the corresponding VaR value at that significance level. Therefore, the CR defined in this study represents the probability that the VaR values are present within a set of generated scenarios. A higher CR value indicates that the generated scenarios are more likely to represent tail risk, indicating that the generated scenarios are more reliable. The formula for the CR (Wang et al. 2017) is

$$C(\tau) = \frac{1}{T} \sum_{t=1}^{T} I\left(P_t^{\min} \le VaR_t(\tau)\right)$$
(23)

$$P_t^{\min} = \min(P_{s,t}), \ s = 1, 2, \dots, S$$
 (24)

where $C(\tau)$ represents the CR with respect to $VaR_t(\tau)$ and $I(\cdot)$ represents an indicator function. $P_{s,t}$ represents the *s*-th scenario value at *t* time and *S* represents the number of scenarios.

	FTSE100		N225		SPX500		DAX	
	CR	сс	CR	сс	CR	сс	CR	СС
$\tau = 0.05$								
GD	0.9762	0.1846	0.9881	0.2427	0.9786	0.2336	0.9762	0.1531
KDE	0.9857	0.1744	0.9929	0.1976	0.9881	0.2408	0.9881	0.1478
MCMC	0.9667	0.2268	0.9691	0.2515	0.9596	0.3002	0.9715	0.1751
LSGAN	1.0000	0.2651	1.0000	0.3147	1.0000	0.3140	1.0000	0.2418
$\tau = 0.01$								
GD	0.9549	0.3328	0.9644	0.3175	0.8955	0.3190	0.9620	0.3077
KDE	0.9881	0.3110	0.9810	0.3407	0.9596	0.3170	0.9834	0.3072
MCMC	0.9335	0.3845	0.9359	0.4695	0.8670	0.4659	0.9216	0.3937
LSGAN	1.0000	0.4339	1.0000	0.6691	1.0000	0.6314	1.0000	0.6006

 Table 11
 Comparision of models using the coverage rate and correlation coefficients

Bold in columns indicate the best method

According to previous studies (Ma et al. 2016; Garatti et al. 2019), we can also use correlation coefficients to measure the similarity in the dynamics between VaR and the generated risk scenarios. We compared the LSGAN model with three commonly used scenario generation methods: the Gaussian distribution (GD), kernel density estimation (KDE), and Markov chain Monte Carlo (MCMC) methods. The evaluation results of the risk scenarios generated using the four methods are listed in Table 11.

Based on the evaluation results presented in Table 11, it is evident that the risk scenarios produced by the LSGAN exhibit a higher probability of encompassing the VaR values and demonstrate a stronger correlation with the VaR dynamics at both significance levels. This observation suggests that utilizing risk scenarios generated by the LSGAN for calculating ES is in line with market risk theory and yields more accurate results than other benchmarks.

Furthermore, as depicted in Fig. 6, the risk scenarios for the four indices were generated using the LSGAN. The 95% probability-level VaR forecasts from QRMogRNNs are denoted by a dashed black line. It can be seen that LSGAN achieves accurate sampling of the tail return distribution beyond VaR, and ensures the diversity and dynamism of the generated risk scenarios. The probability density curves in Fig. 7 depict the distribution of the in-sample 95% VaR data versus the 500 scenarios generated by the pretrained **LSGAN**. By observing the distribution fit, it was found that the LSGAN could learn the characteristics of historical data well and generate reliable synthetic data. The goodnessof-fit between the probability density function (PDF) of the risk scenarios generated by the LSGAN and the PDF of the in-sample VaR values was 0.8882 for FTSE, 0.9298 for N225, 0.8874 for SPX, and 0.9018 for DAX.

Backtesting ES with statistical tests and joint scoring functions

In this section, we verify the performance of the model in terms of ES forecasts, which is a backtesting procedure with six statistical tests and two joint scoring functions. This proved to be effective if the proposed model was successfully tested in more situations. Table 12 presents the p values of the resulting test statistics; the null hypotheses are not rejected in all situations for all desired levels. This shows that the ES estimation approach



Fig. 6 Risk scenarios generated by LSGAN for the four stock market indexes. Gray bands correspond to Chinese stock market crash (2015, 07–2016, 09), 2018 recession (2018, 01–2018, 06), Brexit (2018, 08–2019, 03), and COVID-19 outbreak (2020, 02–2020, 03)



Fig. 7 The distribution of the in-sample 95% VaR data versus 500 scenarios generated by the pre-trained LSGAN

based on LSGAN scenario generation is effective and provides a new tool for estimating the ES in tail risk management.

For comparison with ES forecasts produced by other commonly used scenario generation methods, Table 13 reports the values of the joint scoring functions S_{FZN} (Nolde and Ziegel 2017) and S_{FZ0} (Patton et al. 2019) averaged over the out-of-sample period. A lower scoring function value represents a more accurate ES estimation. The ES

	ES _{UC}	ES _C	MBA	MBR	N&Z	M&F
$\tau = 0.01$					·	
FTSE100	0.9230	0.5030	0.8140	0.8270	1.0000	0.7749
N225	0.8390	0.9074	0.6860	0.7160	0.7252	0.7569
SPX500	0.9240	0.1343	0.4480	0.7270	0.5524	0.7431
DAX	0.5570	0.0589	0.3450	0.2650	0.7252	0.6284
$\tau = 0.05$						
FTSE100	0.9980	0.9850	0.9920	0.9980	1.0000	0.7630
N225	0.9870	0.7430	0.9550	0.9770	0.0846	0.4970
SPX500	1.0000	0.9780	0.9780	0.9970	1.0000	0.8440
DAX	0.8970	0.2640	0.8370	0.8890	0.0846	0.7470
0100	0.0070	0.2040	0.0070	0.0070	0.0040	0.7

Table 12 Out-of-sample ES forecast evaluation by using statistical	tests
--	-------

The bold result indicates that the null hypothesis is not rejected. The hypothesis test form of \mathbf{ES}_{UC} is: $H_0: P_t^{[\alpha]} = F_t^{[\alpha]}, \forall t;$ $H_1: ES_{\alpha,t}^F \ge ES_{\alpha,t'}$ for all t and > for some t, $VaR_{\alpha,t}^F = VaR_{\alpha,t'}$ for all t. The hypothesis test form of \mathbf{ES}_{C} is: $H_0: P_t^{[\alpha]} = F_t^{[\alpha]}, \forall t;$ $H_1: ES_{\alpha,t}^F \ge ES_{\alpha,t'}$ for all t and > for some t, $VaR_{\alpha,t}^F \ge VaR_{\alpha,t'}$ for all t. The hypothesis test form of \mathbf{ES}_{C} is: $H_0: P_t^{[\alpha]} = F_t^{[\alpha]}, \forall t;$ $H_1: ES_{\alpha,t}^F \ge ES_{\alpha,t'}$ for all t and > for some t, $VaR_{\alpha,t}^F \ge VaR_{\alpha,t'}$ for all t.

Table 13 Out-of-sample ES forecast evaluation using joint scoring functions

	FTSE100		N225		SPX500		DAX	
	S _{FZN}	S _{FZ0}						
$\tau = 0.05$								
GD	3.0831	4.8030	2.7255	2.1201	2.9327	3.1874	4.1507	7.6814
KDE	3.1668	4.8842	2.7424	2.1253	2.9239	3.1650	4.2828	8.1125
MCMC	3.1112	4.7895	2.7123	2.1101	2.9078	3.1540	4.1330	7.7470
LSGAN	2.6262	2.2113	2.9775	2.2288	2.6070	1.7495	3.3790	2.6315
$\tau = 0.01$								
GD	4.1411	4.0132	2.6054	1.9298	3.8268	3.4683	3.8268	3.4683
KDE	4.0906	3.9082	2.6791	1.9877	3.6573	3.2680	3.6573	3.2680
MCMC	4.1067	3.9767	2.6171	1.9374	3.7924	3.4298	3.7924	3.4298
LSGAN	3.1220	2.3192	2.7616	1.9758	3.2685	2.4135	3.2918	2.9071

Bold text in columns indicates the best method

estimation process based on the scenario generation of the GD, KDE, and MCMC was the same as that of the LSGAN-based method. The evaluation results of the two joint scoring functions in Table 13 show that the ES estimation results based on the LSGAN scenario generation are more accurate than those based on other scenario generation methods (i.e., GD, KDE, and MCMC).

To offer graphical intuition to support the results, Fig. 8 presents the results of ES outof-sample estimation graphically. By analyzing Fig. 8, we find that the estimated ES is relatively stable during calm periods and larger during periods of turbulent markets, with a better ability to capture extreme risks. The high volatility of ES is obvious during recessions and major economic and financial crises, such as the Chinese stock market crash in 2016, Brexit in 2018, and the COVID-19 outbreak in 2020. Moreover, during the COVID-19 outbreak, the ES values estimated using the proposed model were greater than those after other major adverse events. This shows that the proposed model can



Fig. 8 Out-of-sample forecasts of ES for the four stock market indexes. Gray bands correspond to Chinese stock market crash (2015, 07–2016, 09), 2018 recession (2018, 01–2018, 06), Brexit (2018, 08–2019, 03), and COVID-19 outbreak (2020, 02–2020, 03)

accurately estimate the ES and provide efficient guidance to financial institutions and investors regarding asset allocation.

The training and forecasting of our ES estimation framework do not consume much time. Before out-of-sample forecasting, the LSGAN requires a pretraining process that takes 150 s. In out-of-sample forecasting, we conducted extra training on the LSGAN every ten periods, and the time required to obtain the ES values of the ten periods was 510 s. Therefore, our estimation model can also be used to forecast daily ES.

Summary and conclusions

Finance and economics scholars have long explored better methods for estimating VaR and ES. The structure of the proposed models is often related to the properties of the financial assets. This study follows this research direction and proposes an estimation framework that combines decomposition-aggregation learning with MogRNNs. The estimation framework is consistent with market heterogeneity theory and the properties of asset volatility. MogLSTM and MogGRU can better capture the "long memory" and "clustering" of financial assets by their particular cell structure and the interactive operation between the previous hidden state and the current input. However, these two models have not been extended to predict financial uncertainty. This study proposes combining the above two models with QR to estimate VaR and adds BO to improve practicability. The backtesting results indicate that the model produces reliable VaR estimates. Furthermore, to implement ES estimation, this study proposes a new estimation method using LSGAN to model the distribution of quantiles and generate possible future scenarios of downside risk at a specific probability level. This study also provides a compromise between direct and decomposition forecasts, which balances forecasting accuracy and computational burden. Taking four crucial stock market indices as research objects, five VaR backtesting tests, six ES backtesting tests, and two scoring functions show that the proposed model can forecast risks more accurately than 14 popular benchmarks. We conclude that the proposed model is a promising modeling framework for forecasting risk and is worthy of further study to expand its application.

Abbreviations

CAViaR	Conditional autoregressive value at risk
DQ	Dynamic quantile
ES	Expected shortfall
EMD	Empirical mode decomposition
GANs	Generative adversarial networks
GRU	Gated recurrent unit
HR	Hit ratio
HS	Historical simulation
IMFs	Intrinsic mode functions
LASSO	Least absolute shrinkage and selection operator
LSGAN	Least squares GAN
LSTM	Long short-term memory network
Mog	Mogrifier
MBA	Minimally biased absolute
MBR	Minimally biased relative
QR	Quantile regression
QR-GBDT	Quantile regression gradient boosting decision trees
QRNN	Quantile regression neural network
QR-RF	Quantile regression random forest
QR-SVM	Quantile regression support vector machine
QRMogRNNs	Quantile regression Mogrifier recurrent neural networks
RNNs	Recurrent neural networks
VaR	Value at risk
VMD	Variational mode decomposition

Acknowledgements

We would like to thank the Managing Editor-in-Chief, Professor Gang Kou and the anonymous referees for their constructive comments and suggestions which helped us to improve the manuscript.

Author contributions

JW: conceptualization, supervision. SW: writing-review and editing, software, methodology. ML: validation, visualization. HJ: formal analysis, validation. All authors read and approved the final manuscript.

Funding

This research was supported by the Jiangxi Provincial Natural Science Foundation (20212ACB211003) and the National Natural Science Foundation of China (No. 71671029).

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 26 August 2022 Accepted: 13 November 2023 Published online: 07 January 2024

References

Acerbi C, Szekely B (2014) Backtesting expected shortfall. Risk Mag 27:1-6

- Acerbi C, Szekely B (2017) General properties of backtestable statistics. SSRN Electron J. https://doi.org/10.2139/ssrn. 2905109
- Acerbi C, Tasche D (2002) On the coherence of expected shortfall. J Bank Finance. https://doi.org/10.1016/S0378-4266(02)00283-2
- Artzner P, Delbaen F, Eber JM, Heath D (1999) Coherent measures of risk. Math Finance. https://doi.org/10.1111/1467-9965.00068
- Bertsekas DP (1982) Constrained optimization and Lagrange multiplier methods. Academic Press, New York
- Cai Z, Wang X (2008) Nonparametric estimation of conditional VaR and expected shortfall. J Econom. https://doi.org/10. 1016/j.jeconom.2008.09.005
- Chang YP, Hung MC, Wu YF (2003) Nonparametric estimation for risk in value-at-risk estimator. Commun Stat Part B Simul Comput. https://doi.org/10.1081/SAC-120023877
- Christoffersen PF (1998) Evaluating interval forecasts. Int Econ Rev (philadelphia). https://doi.org/10.2307/2527341 Corsi F (2009) A simple approximate long-memory model of realized volatility. J Financ Econom. https://doi.org/10.1093/ jjfinec/nbp001
- Deng C, Huang Y, Hasan N, Bao Y (2022) Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition. Inf Sci (NY) 607:297–321. https://doi.org/10.1016/J.INS. 2022.05.088

Dragomiretskiy K, Zosso D (2014) Variational mode decomposition. IEEE Trans Signal Process. https://doi.org/10.1109/TSP. 2013.2288675

Du Z, Escanciano JC (2017) Backtesting expected shortfall: accounting for tail risk. Manag Sci. https://doi.org/10.1287/ mnsc.2015.2342

Emenogu NG, Adenomon MO, Nweze NO (2020) On the volatility of daily stock returns of Total Nigeria Plc: evidence from GARCH models, value-at-risk and backtesting. Financ Innov. https://doi.org/10.1186/s40854-020-00178-1

Ener E, Baronyan S, Ali Mengütürk L (2012) Ranking the predictive performances of value-at-risk estimation methods. Int J Forecast. https://doi.org/10.1016/j.ijforecast.2011.10.002

Engle RF, Manganelli S (2004) CAViaR: conditional autoregressive value at risk by regression quantiles. J Bus Econ Stat 22:367–381. https://doi.org/10.1198/073500104000000370

Fatouros G, Makridis G, Kotios D et al (2022) DeepVaR: a framework for portfolio risk assessment leveraging probabilistic deep neural networks. Digit Finance. https://doi.org/10.1007/s42521-022-00050-0

Fissler T, Ziegel JF, Gneiting T (2015) Expected shortfall is jointly elicitable with value at risk—implications for backtesting, pp 1–7

Fu W, Wang K, Tan J, Zhang K (2020) A composite framework coupling multiple feature selection, compound prediction models and novel hybrid swarm optimizer-based synchronization optimization strategy for multi-step ahead shortterm wind speed forecasting. Energy Convers Manag. https://doi.org/10.1016/j.enconman.2019.112461

Garatti S, Ming H, Xie L et al (2019) Scenario-based economic dispatch with uncertain demand response. IEEE Trans Smart Grid. https://doi.org/10.1109/TSG.2017.2778688

Gneiting T (2011) Making and evaluating point forecasts. J Am Stat Assoc. https://doi.org/10.1198/jasa.2011.r10138 Goodfellow JJ, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Advances in neural information processing systems

Grabchak M, Christou E (2021) A note on calculating expected shortfall for discrete time stochastic volatility models. Financ Innov. https://doi.org/10.1186/s40854-021-00254-0

Guo J, Zhao Z, Sun J, Sun S (2022) Multi-perspective crude oil price forecasting with a new decomposition-ensemble framework. Resour Policy 77:102737. https://doi.org/10.1016/J.RESOURPOL.2022.102737

Heusel M, Ramsauer H, Unterthiner T, et al (2017) GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in neural information processing systems

Huang AYH (2013) Value at risk estimation by quantile regression and kernel estimator. Rev Quant Finance Account 41:225–251. https://doi.org/10.1007/s11156-012-0308-x

Huang NE, Shen Z, Long SR et al (1996) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc Lond Ser A Math Phys Eng Sci 454:903–995

Jiang P, Liu Z, Wang J, Zhang L (2021) Decomposition-selection-ensemble forecasting system for energy futures price forecasting based on multi-objective version of chaos game optimization algorithm. Resour Policy 73:102234. https://doi.org/10.1016/j.resourpol.2021.102234

Kim J, Yu J, Kang C et al (2022) A novel hybrid water quality forecast model based on real-time data decomposition and error correction. Process Saf Environ Prot 162:553–565. https://doi.org/10.1016/J.PSEP.2022.04.020

Koenker R, Bassett G (1978) Regression quantiles. Econometrica. https://doi.org/10.2307/1913643 Kupiec PH (1995) Techniques for verifying the accuracy of risk measurement models. J Deriv. https://doi.org/10.3905/jod.

1995.407942 Kwon JH (2021) On the factors of Bitcoin's value at risk. Financ Innov. https://doi.org/10.1186/s40854-021-00297-3

Li Y, Liu Y, Zhu J (2007) Quantile regression in reproducing kernel Hilbert spaces. J Am Stat Assoc 102:255–268. https://doi. org/10.1198/01621450600000979

Li J, Zhou J, Chen B (2020) Review of wind power scenario generation methods for optimal operation of renewable energy systems. Appl Energy. https://doi.org/10.1016/j.apenergy.2020.115992

Liang J, Tang W (2020) Sequence generative adversarial networks for wind power scenario generation. IEEE J Sel Areas Commun. https://doi.org/10.1109/JSAC.2019.2952182

Lopez JA (1999) Methods for evaluating value-at-risk estimates. Econ Rev Fed Reserv Bank San Fran 2:3–17

Ma XY, Sun YZ, Fang HL (2013) Scenario generation of wind power based on statistical uncertainty and variability. IEEE Trans Sustain Energy. https://doi.org/10.1109/TSTE.2013.2256807

Ma R, Xu W, Liu S et al (2016) Asymptotic mean and variance of Gini correlation under contaminated Gaussian model. IEEE Access. https://doi.org/10.1109/ACCESS.2016.2622358

Mao X, Li Q, Xie H, et al (2017) Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, 2017-October, pp 2813–2821. https://doi.org/10.1109/ICCV.2017.304

McNeil AJ, Frey R (2000) Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. J Empir Finance 7:271–300. https://doi.org/10.1016/S0927-5398(00)00012-8

Melis G, Kočiský T, Blunsom P (2019) Mogrifier LSTM, pp 1–13

Meng X, Taylor JW (2020) Estimating value-at-risk and expected shortfall using the intraday low and range data. Eur J Oper Res. https://doi.org/10.1016/j.ejor.2019.07.011

Merlo L, Petrella L, Raponi V (2021) Forecasting VaR and ES using a joint quantile regression and its implications in portfolio allocation. J Bank Financ 133:106248. https://doi.org/10.1016/j.jbankfin.2021.106248

Müller UA, Dacorogna MM, Dave R et al (1993) Fractals and intrinsic time—a challenge to econometricians. Social Science Electronic Publishing, New York

Neshat M, Nezhad MM, Sergiienko NY et al (2022) Wave power forecasting using an effective decomposition-based convolutional bi-directional model with equilibrium Nelder–Mead optimiser. Energy 256:124623. https://doi.org/10. 1016/J.ENERGY.2022.124623

Nguyen LH, Chevapatrakul T, Yao K (2020) Investigating tail-risk dependence in the cryptocurrency markets: a LASSO guantile regression approach. J Empir Finance 58:333–355. https://doi.org/10.1016/j.jempfin.2020.06.006

Nolde N, Ziegel JF (2017) Elicitability and backtesting: perspectives for banking regulation. Ann Appl Stat. https://doi.org/ 10.1214/17-AOAS1041

- Parvini N, Abdollahi M, Seifollahi S, Ahmadian D (2022) Forecasting Bitcoin returns with long short-term memory networks and wavelet decomposition: a comparison of several market determinants. Appl Soft Comput 121:108707. https://doi.org/10.1016/J.ASOC.2022.108707
- Patton AJ, Ziegel JF, Chen R (2019) Dynamic semiparametric models for expected shortfall (and value-at-risk). J Econom. https://doi.org/10.1016/j.jeconom.2018.10.008
- PH H, Rishad A (2020) An empirical examination of investor sentiment and stock market volatility: evidence from India. Financ Innov. https://doi.org/10.1186/s40854-020-00198-x
- Qin T (2020) Stock movement classification from twitter via Mogrifier based memory cells with attention mechanism. In: ACM international conference proceeding series
- Qiu M, Song Y (2016) Predicting the direction of stock market index movement using an optimized artificial neural network model. PLoS ONE. https://doi.org/10.1371/journal.pone.0155133
- Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. J Risk. https://doi.org/10.21314/jor.2000.038
- Takeuchi I, Le QV, Sears TD, Smola AJ (2006) Nonparametric quantile estimation. J Mach Learn Res 7:1231–1264 Taylor JW (2000) A guantile regression neural network approach to estimating the conditional density of multiperiod
- returns. J Forecast. https://doi.org/10.1002/1099-131x(200007)19:4%3c299::aid-for775%3e3.3.co;2-m Taylor JW (2019) Forecasting value at risk and expected shortfall using a semiparametric approach based on the asym-
- metric Laplace distribution. J Bus Econ Stat 37:121–133. https://doi.org/10.1080/07350015.2017.1281815 Wang X, Hu Z, Zhang M, Hu M (2017) Research on establishment of quality evaluation framework of short-term wind
- power scenarios. Dianwang Jishu/power Syst Technol. https://doi.org/10.13335/j.1000-3673.pst.2016.1985 Wang K, Fu W, Chen T et al (2020) A compound framework for wind speed forecasting based on comprehensive feature
- selection, quantile regression incorporated into convolutional simplified long short-term memory network and residual error correction. Energy Convers Manag 222:113234. https://doi.org/10.1016/j.enconman.2020.113234
- Wang J, Cui Q, Sun X, He M (2022a) Asian stock markets closing index forecast based on secondary decomposition, multifactor analysis and attention-based LSTM model. Eng Appl Artif Intell 113:104908. https://doi.org/10.1016/J.ENGAP PAI.2022.104908
- Wang J, Wang S, Zeng B, Lu H (2022b) A novel ensemble probabilistic forecasting system for uncertainty in wind speed. Appl Energy 313:118796. https://doi.org/10.1016/j.apenergy.2022.118796
- Wang J, Zhang L, Liu Z, Niu X (2022c) A novel decomposition-ensemble forecasting system for dynamic dispatching of smart grid with sub-model selection and intelligent optimization. Expert Syst Appl 201:117201. https://doi.org/10. 1016/J.ESWA.2022.117201
- Wang Z, Li H, Chen H et al (2022d) Linear and nonlinear framework for interval-valued PM2.5 concentration forecasting based on multi-factor interval division strategy and bivariate empirical mode decomposition. Expert Syst Appl 205:117707. https://doi.org/10.1016/J.ESWA.2022.117707
- White H (1992) Nonparametric estimation of conditional quantiles using neural networks. In: Computing science and statistics
- Xu Q, Jiang C, He Y (2016) An exponentially weighted quantile regression via SVM with application to estimating multiperiod VaR. Stat Methods Appl 25:285–320. https://doi.org/10.1007/s10260-015-0332-9
- Yuan R, Wang B, Mao Z, Watada J (2021) Multi-objective wind power scenario forecasting based on PG-GAN. Energy. https://doi.org/10.1016/j.energy.2021.120379
- Zhang X, Lai KK, Wang SY (2008) A new approach for crude oil price analysis based on empirical mode decomposition. Energy Econ. https://doi.org/10.1016/j.eneco.2007.02.012
- Zhu Y, Mariani G, Li J (2021) Pagan: portfolio analysis with generative adversarial networks. SSRN Electron J. https://doi. org/10.2139/ssrn.3755355
- Žiković S, Filer RK (2013) Ranking of VaR and ES models: performance in developed and emerging markets. Financ a Uver - Czech J Econ Financ. https://doi.org/10.2139/ssrn.2171673

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- ► Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at > springeropen.com