Machine Learning Based Portfolio Selection Under Systemic Risk

Weidong Lin[1] and Abderrahim Taamouti[2*]

[1]Finance Department, NEOMA Business School, Rouen Campus, 1 Rue du Maréchal Juin, Mont-Saint-Aignan, 76130, France.
[2*]Management School, University of Liverpool, Chatham St, Liverpool, L69 7ZH, UK.

*Corresponding author. E-mail: Abderrahim.Taamouti@liverpool.ac.uk;

Abstract

This paper aims to enhance the classical mean-variance portfolio selection by using machine learning techniques and accounting for systemic risk. The optimal portfolio is solved through a three-step supervised learning model. Firstly, the Smooth Pinball Neural Network is employed to predict return distributions of individual assets and the market. Secondly, we use copula to model dependence between assets and the market, based on which we simulate return scenarios. Lastly, we maximize an ex-ante conditional Sharpe ratio conditioning on systemic events. We run a large-scale comparative study using nearly 600 US individual stocks over 37 years. Our set of predictors includes 94 firm characteristics, 14 macroeconomic variables, and 74 industry dummies. The backtesting results demonstrate the superiority of our proposed approach over popular benchmark strategies including a GARCH-based model. This outperformance is statistically significant and robust to the inclusion of transaction costs.

Keywords: machine learning, portfolio selection, systemic risk, simulation, probabilistic forecasting

JEL Classification: G11 , C45 , C53 , C58

1 Introduction

Machine learning (ML) can be used to explicitly describe complex relationships and uncover patterns within high-dimensional datasets. In the empirical finance literature, Gu et al. (2020) and others narrowed the definition of ML down to a set of high-dimensional statistical prediction models, combined with optimization algorithms for parameter searching and regularization methods for overfitting mitigation. In this paper, we combine ML and a novel methodology to build optimal portfolios that deal with systemic risk. The new approach enables us to incorporate information from high-dimensional datasets and account for systemic events simultaneously.

Stock return predictability is of great importance to investors as it is a key ingredient for asset allocation, risk management, asset pricing, etc. Many studies try to explain cross-sectional stock returns using various predictors such as size, book-to-market, and momentum factor; see Harvey et al. (2016) and references therein. The increasing number of available factors might provide richer predictive information by incorporating big data into stock return modelling. However, as argued by Gu et al. (2020) and others, traditional prediction methods (e.g. linear models) are unable to fit complex patterns and tend to break down when the number of covariates is close to the number of observations or when the predictors are highly correlated. Thus, thanks to its ability to handle high-dimensional datasets and complex nonlinear relationships, ML is the tool we need to confront the challenge of improving prediction accuracy and consequently portfolio performance.

Since the ML techniques have shown promising superiority against traditional statistical methods in stock return prediction, many researchers have applied these models to portfolio optimization and generated satisfying results; see Zhang et al. (2020); Babiak and Baruník (2020); and Huang et al. (2021); among others. However, as far as we know, the existing literature has not yet explored the potential economic gains of combining ML-based probabilistic return forecasts with portfolio optimization. The applications in FinTech focus mostly on point forecasts of stock returns without accounting for any uncertainty. Moreover, so far the efficiency of portfolios constructed using ML techniques has been tested mainly for characteristic-sorted portfolios (e.g. long-short decile portfolios), which further motivates us to investigate whether the ML approach to probabilistic forecasting will help our investors when forming optimal portfolios.

Starting from the mean-variance paradigm of Markowitz (1952), the tradeoff between return and risk has become the research focus in portfolio optimization. A general approach for building optimal portfolios is to maximize an ex-ante reward-risk ratio that is based on diverse perceptions of reward and risk measures; see Rachev et al. (2008). Although the existing performance ratios have led to the development of many major theories and practices on optimal asset allocations, by construction, they are unable to take into account systemic risk that might affect the portfolio's risk beyond the effect of individual assets' risks. In other words, these ratios only focus on measuring the performance of portfolio assets without accounting for systemic events happening in the market. Hence, investors need new portfolio strategies that can help them overcome systemic risk during their decision-making process. Loosely speaking, systemic risk can be defined as the risk of collapse of the whole financial system, as opposed to the risk associated with any individual entity of the system (Lin et al. 2023). The global financial crisis of 2007-2008 and the subsequent crises (e.g. euro debt crisis and COVID-19 pandemic) are consequences of ignoring this kind of risk.

While the finance literature has proposed various systemic risk measures, there is scarce research on portfolio management under systemic risk. Only a few papers examined the implications of systemic risk on investment decisions; see Capponi and Rubtsov (2022) and references therein. Recently, Lin et al. (2023) were interested in solving the tradeoff between reward and risk under stressed market conditions by introducing a conditional Sharpe ratio (CoSR), in which they incorporate the occurrence of systemic events into the performance measure. However, none of the above-mentioned papers utilizes ML techniques for predicting returns when building optimal portfolios. The present paper bridges this gap by merging the literature on portfolio selection under systemic risk with the one on cross-sectional return prediction using ML.

Theoretically, solving the portfolio optimization problem by maximizing a specific performance ratio requires knowing the true distribution of future portfolio returns. In practice, however, we might just need to estimate the reward and risk measures to solve the portfolio problem. The estimation of these measures can be done in two different ways that involve either using historical observations of return or simulated return scenarios. It has been argued that the optimal portfolios obtained based on the former approach are unlikely to beat the naive portfolio, which can be mainly attributed to their extreme weights over the out-of-sample period; see DeMiguel et al. (2009) among others. This might be caused by the estimation errors that are known to affect sample-based estimators and make the latter less effective when they are used as inputs in an optimization problem. Therefore, it is important to find ways to robustify these portfolio optimization inputs.

Although extensive effort has been devoted to alleviating the aforementioned problem by developing estimators that take into account estimation errors; see for example Branger et al. (2019), few papers in the literature resort to promising ML tools. In this paper, we fill this gap by using a distributional ML approach to generate return scenarios that we use to estimate the reward and risk measures. Specifically, we formulate the portfolio selection problem as a three-stage supervised learning process that considers systemic risk when building optimal portfolios. We start by predicting quantiles of cross-sectional stock returns using the smooth pinball neural network (SPNN) model, based on which we estimate the conditional marginal distributions of returns on portfolio assets and the market. Thereafter, we apply t-copula to model the dependence between individual assets and the market, and generate scenarios for future returns. Lastly, we solve the portfolio optimization problem dynamically by maximizing CoSR based on the simulated return scenarios.

Furthermore, we perform a large-scale empirical study using nearly 600 US stocks with 37 years of history from 1985 to 2021. Our set of predictors includes 94 firm characteristics, 14 macroeconomic variables, and 74 industry dummies. For the returns of individual assets and the market, we calculate their monthly quantile forecasts using SPNN. Thereafter, on each month within our out-of-sample period, we solve the portfolio optimization problem dynamically using different objective functions. Specifically, we feed the CoSR optimizers with input parameters that we estimate using the return scenarios generated from a hybrid model that combines SPNN and copula. For comparison, we calculate sample-based tangency portfolio (SR), sample-based global minimum variance portfolio (GMV), and equally weighted portfolio (1/N) as benchmark strategies. In addition, we also consider an alternative scenario-based method proposed by Lin et al. (2023) as a strong competitor, in which the authors implemented a GARCH-copula hybrid model to simulate returns. Finally, we calculate and report the out-of-sample portfolio performance of different strategies via a backtesting analysis. We also test the significance of the difference in Sharpe ratios between our approach and that of each benchmark portfolio.

Our paper contributes to the literature in two ways. First, we shed new light on portfolio selection using distributional ML approaches. This is done by incorporating SPNN-based probabilistic forecasts of stock returns into a conditional Sharpe ratio. The ability of ML methods to capture complex and nonlinear patterns that characterize big datasets results in more accurate forecasts of future return distributions, and hence in more robust estimates that are then fed into the portfolio optimizer, which in turn enhances the portfolio performance relative to popular benchmark portfolios. Second, we further improve the portfolio selection process by explicitly incorporating systemic events, which leads to portfolios that are resilient during crises. After accounting for the conditional tail risk of portfolios, our backtesting results show that our proposed approach not only performs well on stressed scenarios but also performs steadily when the market is doing well.

The rest of the paper is organized as follows. Section 2 introduces the stock return quantile prediction via SPNN. Section 3 formulates the portfolio selection problem under systemic risk. In this same section, we discuss the method of probabilistic forecasting of returns, and illustrate how we model dependence through copula and generate return scenarios. Section 4 uses a high-dimensional dataset from the US market to conduct a large-scale empirical analysis in which we compare the out-of-sample portfolio performance of our proposed approach with those of several popular benchmark strategies. Section 5 concludes. Figures and tables that are not shown in the body of the paper are in Appendix A and B, respectively.

2 Quantile regression neural network

We start by reviewing the traditional quantile regression (QR), which is the main building block of the Quantile Regression Neural Network (QRNN). Then we introduce the mathematical formulation of QRNN and its advanced variant namely the Smooth Pinball Neural Network (SPNN). Before we describe those quantile models, let us first set some notations. Using the terminology of the literature on neural networks, we denote by $\boldsymbol{R} = (R_1, ..., R_V)$ the $1 \times V$ vector of monthly returns for V training samples, and $\boldsymbol{X} = (\boldsymbol{X}_1, ..., \boldsymbol{X}_V)$, with $\boldsymbol{X}_v = (x_{1,v}, ..., x_{P,v})^T$, for $v = 1, ..., V$, the $P \times V$ matrix of P covariates across V training samples, including firm-level features, interactions of each feature with macroeconomic variables, and industry dummies.[1]

[1] Note that in the above notations, we do not use any subscript to distinguish between different entities (e.g. individual firms), but we will do so in Section 3.

2.1 Model specification

Initially proposed by Koenker and Bassett (1978), the quantile regression (QR) model estimates the relationship between predictors and a conditional quantile of the response variable. Formally, the τ-th conditional quantile of the predictand R_v is given by

$$Q_{R_v}(\tau|\boldsymbol{X}_v) = \boldsymbol{X}_v^T \boldsymbol{\beta}(\tau), \quad \forall v \in \{1, ..., V\}, \quad \tau \in (0,1), \tag{1}$$

where $\boldsymbol{\beta}(\tau) = [\beta_0(\tau), ..., \beta_P(\tau)]^T$ is the vector of the regression coefficients and can be estimated by solving the following optimization problem

$$\hat{\boldsymbol{\beta}}(\tau) = Arg \min_{\boldsymbol{\beta}(\tau)} \frac{1}{V} \sum_{v=1}^{V} \rho_\tau \left[R_v - \boldsymbol{X}_v^T \boldsymbol{\beta}(\tau)\right], \tag{2}$$

where the asymmetric loss function ρ_τ (known as pinball loss function) is defined as

$$\rho_\tau(u) = \begin{cases} \tau u & u \geq 0 \\ (\tau - 1)u & u < 0 \end{cases}. \tag{3}$$

The fitted conditional quantile is expressed as $\hat{Q}_{R_v}(\tau|\boldsymbol{X}_v) = \boldsymbol{X}_v^T \hat{\boldsymbol{\beta}}(\tau)$.

QR provides a more complete picture of the conditional distribution of $\boldsymbol{R}$ than conditional mean regression and does not make assumptions on the distribution of the target variable. Moreover, QR is robust to outliers and can thus be estimated more accurately than conventional moments regression. The QR model defined in (1) is, however, unable to capture possible nonlinear relationships between $\boldsymbol{R}$ and $\boldsymbol{X}$. To overcome this issue, Taylor (2000) originally introduced the quantile regression neural network (QRNN) that combines QR with ANN to depict the complex nonlinear relationships between predictors and the response variable without pre-specifying a functional form. Thus, instead of using a linear function, the conditional quantile is approximated by a neural network $f(\cdot)$ under the QRNN framework. Formally, the conditional τ-th quantile of R_v based on the QRNN model with a single hidden layer can be formulated as

$$Q_{R_v}(\tau|\boldsymbol{X}_v) = f\left(\boldsymbol{X}_v, \boldsymbol{H}(\tau), \boldsymbol{O}(\tau)\right) = g_2\Big[\sum_{k=1}^{K} o_k(\tau) g_1\big(\sum_{j=1}^{P} h_{j,k}(\tau) x_j^v\big)\Big], \tag{4}$$

where $\boldsymbol{H}(\tau) = (h_{1,1}(\tau), ..., h_{P,K}(\tau))^T$ is the weight vector that links the input and hidden layer, $\boldsymbol{O}(\tau) = (o_1(\tau), ..., o_K(\tau))^T$ is the weight vector responsible for connecting the hidden and output layer, and K is the number of hidden neurons. The activation functions $g_1(\cdot)$ and $g_2(\cdot)$ are generally specified as a sigmoid/rectifier function and a linear function, respectively. The set of parameters $\boldsymbol{\beta}(\tau) \equiv \big\{\boldsymbol{H}(\tau), \boldsymbol{O}(\tau)\big\}$ can be estimated by solving

$$\hat{\boldsymbol{\beta}}(\tau) = \underset{\boldsymbol{\beta}(\tau)}{\text{Argmin}}\, L(\tau) = \underset{\boldsymbol{\beta}(\tau)}{\text{Argmin}} \frac{1}{V} \sum_{v=1}^{V} \rho_\tau \Big[\big(R_v - f(\boldsymbol{X}_v, \boldsymbol{\beta}(\tau))\big)\Big], \tag{5}$$

and the fitted conditional quantiles are obtained as $\hat{\boldsymbol{Q}}_{\boldsymbol{R}}(\tau|\boldsymbol{X}) = f(\boldsymbol{X}, \hat{\boldsymbol{\beta}}(\tau))$. Figure 1 illustrates the architecture of a QRNN model with a single hidden layer.

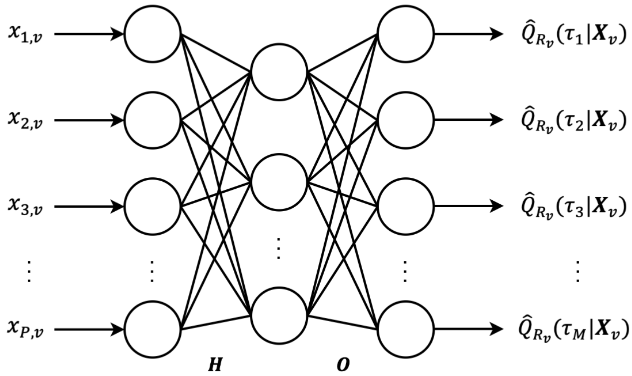


Fig. 1: Schematic diagram of SPNN with a single hidden layer.

2.2 Approximation of pinball loss function

The parameters of neural networks are typically determined through some gradient-based nonlinear optimization algorithms by which the gradients are calculated using the backpropagation algorithm; see Cannon (2011). The gradient of (5) can be computed analytically by updating backpropagation equations based on the least absolute error function. However, the loss function ρ_τ is non-differentiable at the origin ($u = 0$), which thus requests a smooth approximation in order to apply gradient-based optimization methods. To smooth ρ_τ, one can resort to the Huber norm introduced by Huber (2004), which is defined as:

$$h(u) = \begin{cases} \frac{1}{2}u^2 & |u| \leq \varepsilon \\ \varepsilon(|u| - \frac{1}{2}\varepsilon) & \text{otherwise} \end{cases}, \tag{6}$$

where ε is a given threshold magnitude; see also Cannon (2018) and Xu et al. (2017). The check function is approximated by

$$\rho_\tau^{(A)}(u) = |\tau - I_{\{u<0\}}|\, h(u), \tag{7}$$

where $I_{\{u<0\}}$ is an indicator function that values as one when $u < 0$ and zero otherwise. As ε converges to zero, the approximate error function converges to the exact QR error function; see Xu et al. (2017). An alternative way to smooth the loss function was proposed by Zheng (2011), which smoothes ρ_τ using a logistic function, i.e., for $\tau \in (0, 1)$,

$$\rho_\tau^{(A)}(u) = \tau u + \alpha \ln\big(1 + \exp(-\frac{u}{\alpha})\big), \tag{8}$$

where $\alpha > 0$ is the smoothing parameter. As argued by Arends et al. (2020), the loss function in equation (8) combines Huber loss and pinball loss together. Zheng (2011) has shown that $\rho_\tau^{(A)}(u) = \rho_\tau(u)$ as $\alpha \to 0^+$ in the limit. For illustration, Figure 2 below displays the pinball loss function (red curve) for $\tau = 0.5$, the Huber approximation (blue curve) for $\varepsilon = 10$, and the logistic approximation (black curve) for $\alpha = 0.2$, respectively.

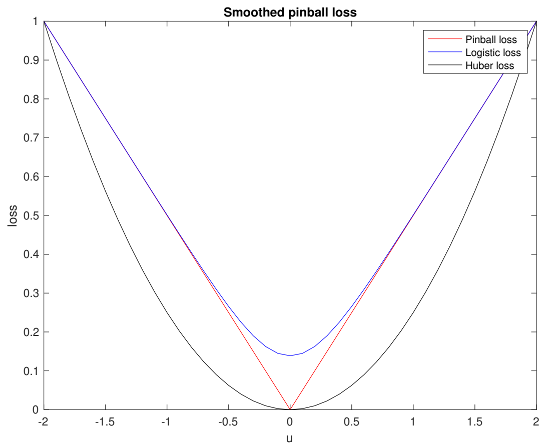


Fig. 2: Pinball loss versus smoothed ones.

Replacing the loss function in equation (5) by a smoothed $\rho_\tau^{(A)}$, we obtain the following updated objective function

$$L^{(A)}(\tau) = \frac{1}{V} \sum_{v=1}^{V} \rho_\tau^{(A)} \Big[\big(R_v - f(\boldsymbol{X}_v, \boldsymbol{\beta}(\tau)) \big) \Big]. \tag{9}$$

We can minimize (9) using standard gradient-based optimization algorithms to obtain the estimate of $\boldsymbol{\beta}(\tau)$. Cannon (2011) implemented this optimization procedure in R using the quasi-Newton optimization algorithm for calculating the Huber loss, while Hatalis et al. (2019) applied the logistic loss in Python. We adopt the logistic loss (8) in our empirical analysis.[2]

2.3 Smooth pinball neural network

To further enhance the performance of estimating quantiles, Xu et al. (2017) extended the original QRNN model to composite quantile regression neural network (CQRNN), which can be used to estimate multiple conditional quantiles (for different values of

[2]We have also tried for the Huber loss and the backtesting results are similar to those of using logistic loss.

τ) simultaneously and efficiently. CQRNN shares the same goal as the one of linear composite quantile regression (CQR) developed by Zou and Yuan (2008), namely combining the strength across multiple quantile regressions to better capture the complex nonlinear relationships between the predictors and the predictand (Cannon 2018). CQRNN is similar to QRNN by structure, where the difference lies in the objective function, which is now summed over M values of τ:

$$L_C^{(A)} = \frac{1}{M} \sum_{m=1}^{M} L^{(A)}(\tau_m), \tag{10}$$

where τ is equally spaced as $\tau_m = \frac{m}{M+1}$ for $m \in \{1, \cdots, M\}$. The expression in (10) is a composite version of the objective function in equation (9) since it evaluates multiple regression quantiles synthetically. CQRNN is a flexible model not only because it allows us to uncover complex nonlinear patterns among variables through the properties of ANN, but also because it helps improve the estimation efficiency and prediction accuracy thanks to the property of CQR (Xu et al. 2017).

Although CQRNN improves the model efficiency and prediction accuracy, it fails to prevent quantile crossing. Quantile crossing violates the property that a cumulative distribution function (CDF) is monotonically increasing. As stated by Ouali et al. (2016), it is a serious modelling problem that may result in an invalid predictive distribution of the predictand. Similarly, Bang et al. (2016) argued that this problem reduces the estimation accuracy of regression quantiles and can cause trouble to the subsequent analysis and interpretation of the model. In order to mitigate this issue, Cannon (2018) developed a monotonic CQRNN (MCQRNN) model that imposes partial monotonicity constraints on the neural network weights and stacks covariates into an input matrix. MCQRNN imposes monotonicity constraints on a standard Multi-Layer Perceptron (MLP) and then integrates the model architecture of CQRNN to achieve simultaneous estimation. However, the stacked matrix of covariates complicates the network by adding overmuch parameters, which makes the estimation computationally inefficient and induces the propensity of overfitting.

Recently, Hatalis et al. (2019) proposed an efficient alternative to MCQRNN namely smooth pinball neural network (SPNN) that introduces a set of constraints into the CQRNN framework. To prevent the quantile crossover, the constraint $Q_{R_v}(\tau_1|\boldsymbol{X}_v) \leq \cdots \leq Q_{R_v}(\tau_M|\boldsymbol{X}_v)$, $\forall v$, needs to be satisfied. However, it is hard to solve the optimization problem via gradient-based methods with such constraints. To fix this issue, Hatalis et al. (2019) suggested adding a penalty term to the objective function (10), where the penalty term p is defined as

$$p = c\frac{1}{MV} \sum_{m=1}^{M} \sum_{v=1}^{V} \Big[\max\Big(0, \epsilon - \big(\hat{Q}_{R_v}(\tau_m|\boldsymbol{X}_v) - \hat{Q}_{R_v}(\tau_{m-1}|\boldsymbol{X}_v)\big)\Big)\Big]^2, \tag{11}$$

where $\hat{Q}_{R_v}(\tau_0|\boldsymbol{X}_v)$ is initialized to zero, ϵ denotes the minimum difference value between two neighbouring quantiles, and c is the penalty parameter. The objective function of SPNN is given by

$$L_S = L_C^{(A)} + p + \lambda||\boldsymbol{\beta}||_1, \tag{12}$$

where $\boldsymbol{\beta} \equiv \{\boldsymbol{H}, \boldsymbol{O}\} = \{\boldsymbol{H}(\tau_m), \boldsymbol{O}(\tau_m)\}_{m=1,...,M}$ represents the composite parameters of neural network (i.e. parameters across all values of τ). Note that the l_1 norm $||\cdot||_1$ is applied in (12) to mitigate the overfitting problem, where λ denotes the regularization parameter. The training of SPNN can be conducted using standard gradient-based optimization algorithms. In our paper, we adopt SPNN for completing prediction tasks due to its virtues of simultaneously estimating multiple quantiles and preventing quantile crossing.

3 Portfolio selection under systemic risk

In this section, we first review the CoSR-based portfolio selection problem. Then we discuss the simulation scheme for generating multivariate return scenarios, which is done by combining the predicted conditional marginal return densities from SPNN with a fitted t-copula. Lastly, we show how to calculate the CoSR estimator based on simulated returns.

3.1 Portfolio selection problem

We consider an economy with N risky assets. Hereafter, we present the portfolio allocation problem of an investor that wishes to select the weights of N assets by maximizing an ex-ante CoSR measure following Lin et al. (2023). Before we describe our portfolio problem, let us first set some notations. We denote by $\boldsymbol{R}_t = (R_{1,t}, ..., R_{N,t})^T$ the vector of monthly returns over month t, $R_{m,t}$ the market return over month t, and $\boldsymbol{W}_t = (\omega_{1,t}, ..., \omega_{N,t})^T$ the vector of portfolio weights held over month $t+1$. The portfolio return is given by $R_{p,t+1} = \boldsymbol{W}_t^T \boldsymbol{R}_{t+1}$. $\mathbf{0}$ and $\mathbf{1}$ denote the column vectors of zeros and ones, respectively.

A generic portfolio optimization problem when an investor's objective function is given by a performance measure $\phi(\cdot)$ can be described as follows

$$\boldsymbol{W}_t^* = \arg\max_{\boldsymbol{W}_t} \phi_t(R_{p,t+1}), \quad \text{s.t. } \mathbf{1}^T \boldsymbol{W}_t = 1, \tag{13}$$

where the different candidates of $\phi(\cdot)$ result in different optimal portfolios. As we argued in previous sections, we are interested in building portfolios that take into account systemic risk. For this reason, we consider the performance measure CoSR that is defined as a ratio of a conditional reward measure over a conditional risk measure

$$\text{CoSR}_t(R_{p,t+1}) = \frac{\text{CoER}_t(R_{p,t+1})}{\text{CoSD}_t(R_{p,t+1})}, \tag{14}$$

with the conditional reward CoER defined as

$$\text{CoER}_t(R_{p,t+1}) = E_t(R_{p,t+1} - R_{m,t+1}|\text{SE}_{t+1}) = \boldsymbol{W}_t^T \boldsymbol{\mu}_{t|\text{SE}} - \mu_{m,t|\text{SE}}, \tag{15}$$

where $\text{SE}_{t+1} = \{R_{m,t+1} < C\}$ denotes a systemic event (SE) where the market return goes below a certain threshold C over the next month, $\boldsymbol{\mu}_{t|\text{SE}} = E_t(\boldsymbol{R}_{t+1}|\text{SE}_{t+1})$ is the vector of conditional expected returns on individual assets, and $\mu_{m,t|\text{SE}} = E_t(R_{m,t+1}|\text{SE}_{t+1})$ is the conditional expected market return. Analogously, the conditional risk measure CoSD is defined as the conditional second moment of the portfolio's excess return, that is:

$$
\begin{aligned}
\text{CoSD}_t(R_{p,t+1}) &= \left[Var_t(R_{p,t+1} - R_{m,t+1}|\text{SE}_{t+1})\right]^{1/2} \\
&= \left(\boldsymbol{W}_t^T \boldsymbol{\Sigma}_{t|\text{SE}} \boldsymbol{W}_t + \sigma^2_{m,t|\text{SE}} - 2\boldsymbol{W}_t^T \boldsymbol{\sigma}_{t|\text{SE}}\right)^{1/2},
\end{aligned} \tag{16}
$$

where $\boldsymbol{\Sigma}_{t|\text{SE}} = Var_t(\boldsymbol{R}_{t+1}|\text{SE}_{t+1})$ denotes the conditional covariance matrix of asset returns, $\sigma^2_{m,t|\text{SE}} = Var_t(R_{m,t+1}|\text{SE}_{t+1})$ denotes the conditional variance of market return, and $\boldsymbol{\sigma}_{t|\text{SE}} = cov_t(\boldsymbol{R}_{t+1}, R_{m,t+1}|\text{SE}_{t+1})$ is the vector of conditional covariances between individual assets and the market portfolio. The portfolio selection problem under CoSR is given by

$$
\boldsymbol{W}_t^* = \arg\max_{\boldsymbol{W}_t} \left\{\text{CoSR}_t(R_{p,t+1})\right\}, \quad \text{s.t. } \mathbf{1}^T \boldsymbol{W}_t = 1. \tag{17}
$$

As pointed out by Lin et al. (2023), the optimization problem in (17) can be solved analytically without imposing short-selling constraints ($\boldsymbol{W} \geq \mathbf{0}$). However, investors might benefit from prohibiting short sales in certain cases. For example, financial regulators often ban short-selling to restrain short-term speculative investments, which is also realistic in extreme scenarios. Furthermore, the capacity of short-selling tends to produce extreme portfolio weights during market distress, thus increasing the transaction costs triggered by portfolio rebalancing. Hence, we consider no short-sale constraint in our later exercise. Unfortunately, (17) has no closed-form solution under short-selling restrictions. Thus, we employ a numerical procedure to solve it.

3.2 Simulation of return scenarios

To solve (17), we adopt a scenario-based method via Monte Carlo simulation. The proposed performance ratio can be estimated using its empirical analogue based on the generated return scenarios over the subset of crisis scenarios.

In this section, we discuss how we estimate the conditional marginal distributions (densities) of monthly returns. In particular, we consider a nonparametric estimation approach for predictive densities using conditional quantiles obtained from SPNN models. After fitting the marginal densities, we apply t-copula to model the dependence between assets and market returns. Lastly, we describe an algorithm for simulating return scenarios.

3.2.1 Predictive densities from quantiles

Let $\boldsymbol{X}_{j,t} = \{x_{j,p,t}\}_{p=1,...,P;\, t=1,...,T}$ for $j \in \{i, m\}$ with $i = 1, ..., N$ be the P-dimensional predictor set for monthly return of firm i or market index, which is available at the end of the month t. Hereafter, we show how the conditional quantiles of returns obtained from SPNN models, i.e. $\hat{q}_{j,t+1}(\tau_m) = \hat{Q}_{R_{j,t+1}}(\tau_m|\boldsymbol{X}_{j,t})$, can be used to estimate the conditional density $p_{j,t} = p(R_{j,t+1}|\boldsymbol{X}_{j,t})$ following Cannon (2011). Formally, to recover the predictive probability density $\hat{p}_{j,t}(\cdot)$ based on conditional quantiles, we distinguish between the following three cases:

- If $\hat{q}_{j,t+1}(\tau_1) \le R_{j,t+1} < \hat{q}_{j,t+1}(\tau_M)$ and τ_m and τ_{m+1} are such that $\hat{q}_{j,t+1}(\tau_m) \le R_{j,t+1} < \hat{q}_{j,t+1}(\tau_{m+1})$, then

$$\hat{p}_{j,t} = \frac{\tau_{m+1} - \tau_m}{\hat{q}_{j,t+1}(\tau_{m+1}) - \hat{q}_{j,t+1}(\tau_m)}. \tag{18}$$

- If $R_{j,t+1} < \hat{q}_{j,t+1}(\tau_1)$, we assume a lower exponential tail

$$\hat{p}_{j,t} = z_1 \exp\Big(-\frac{|R_{j,t+1} - \hat{q}_{j,t+1}(\tau_1)|}{e_1}\Big), \tag{19}$$

  where $z_1 = (\tau_2 - \tau_1)/(\hat{q}_{j,t+1}(\tau_2) - \hat{q}_{j,t+1}(\tau_1))$ and $e_1 = \tau_1/z_1$.
- If $R_{j,t+1} \ge \hat{q}_{j,t+1}(\tau_M)$, we assume an upper exponential tail

$$\hat{p}_{j,t} = z_M \exp\Big(-\frac{|R_{j,t+1} - \hat{q}_{j,t+1}(\tau_M)|}{e_M}\Big), \tag{20}$$

  where $z_M = (\tau_M - \tau_{M-1})/(\hat{q}_{j,t+1}(\tau_M) - \hat{q}_{j,t+1}(\tau_{M-1}))$ and $e_M = \tau_M/z_M$.

As illustrated above, the predictive densities are interpolated between neighboring quantiles, while the exponentially decaying tails ensure the probability density function integrates to one.

3.2.2 Dependence modelling and scenario generation

Once the predictive marginal return distributions for individual assets and the market are obtained, the next is to model joint return distribution via copula. An $(N+1)$-dimensional copula C is a multivariate distribution function on $[0,1]^{N+1}$, with standard uniform margins. Following Sklar's theorem (Sklar 1959), any multivariate distribution, in our case the multivariate distribution function of individual firm and market monthly returns, can be decomposed into univariate margins and a certain copula function, that is

$$F_{R_1,\ldots,R_{N+1}}(u_1, ..., u_{N+1}) = C\left(F_{R_1}(u_1), ..., F_{R_{N+1}}(u_{N+1})\right), \tag{21}$$

where $u_j \sim U(0,1)$ for $j = 1, ..., N+1$, $R_{N+1} = R_m$, and F_{R_j} denotes the marginal CDF of monthly return on an individual asset or market index.

In our empirical analysis, we use t-copula to model the dependence among monthly returns. The t-copula function is given by

$$C_{\nu,\boldsymbol{\mathcal{P}}}(u_1, ..., u_{N+1}) = \int_{-\infty}^{t_\nu^{-1}(u_1)} \cdots \int_{-\infty}^{t_\nu^{-1}(u_{N+1})} \frac{\Gamma(\frac{\nu+N+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{(\nu\pi)^{N+1}|\boldsymbol{\mathcal{P}}|}} \Big(1 + \frac{x'\boldsymbol{\mathcal{P}}^{-1}x}{\nu}\Big)^{-\frac{\nu+N+1}{2}} dx, \tag{22}$$

where Γ is the Gamma function, $\boldsymbol{\mathcal{P}}$ is a correlation matrix, and ν represents the degrees of freedom. We generate return scenarios according to the following steps:

- Given historical monthly returns on firms and market, i.e, $\{R_{j,t}\}_{j=1,...,N+1;\, t=1,...,T}$, we estimate the CDF, say $\hat{F}_{\nu_{j,t}}$, of return series $\{R_{j,t}\}$ using a univariate t-location-scale distribution, i.e. $R_{j,t} \sim \hat{F}_{\nu_{j,t}}$.
- Convert historical monthly returns over each estimation window into standard uniforms using probability transformation: $u_{j,t} = \hat{F}_{\nu_{j,t}}(R_{j,t})$, where $u_{j,t} \sim U(0,1)$.
- Given $\{u_{j,t}\}_{j=1,...,N+1}$, we use the method of moment to estimate the degrees of freedom ν and the correlation matrix $\boldsymbol{\mathcal{P}}$ of the t-copula; see McNeil et al. (2015).
- Simulate dependent standard uniform vectors $\boldsymbol{u}_{t+1}^{(s)} = \left(u_{1,t+1}^{(s)}, \cdots u_{N+1,t+1}^{(s)}\right)$ for $s = 1, ..., S$, where S is the simulation sample size.
- Convert $\boldsymbol{u}_{t+1}^{(s)}$ to return scenarios via quantile transformation: $R_{j,t+1}^{(s)} = \hat{F}_{R_{j,t+1}}^{-1}(u_{j,t+1}^{(s)})$, where $\hat{F}_{R_{j,t+1}}^{-1}$ is the inverse CDF of the fitted j-th marginal empirical distribution deduced from $\hat{p}_{j,t}$ for $j \in \{i, m\}$. From this, we obtain S simulated return samples over month $t+1$ that possess the same dependence structure as the in-sample dataset.

3.3 CoSR estimation

To estimate the performance measure CoSR based on simulated returns, we first estimate the elements of the vector of conditional expected returns on individual assets $\mu_{t|\mathrm{SE}}$ using the average of the simulated arithmetic asset returns over one-month ahead period, that is

$$\hat{\mu}_{i,t|\mathrm{SE}} = \frac{\sum_{s=1}^{S} R_{i,t+1}^{(s)}\, I\{R_{m,t+1}^{(s)} < C\}}{\#\mathrm{SE}}, \tag{23}$$

where S is the number of Monte Carlo simulations and $\#\mathrm{SE} = \sum_{s=1}^{S} I\{R_{m,t+1}^{(s)} < C\}$ is the number of scenarios out of S that represent a market distress. For each asset in the portfolio, the filtered mean vector (average of one-period ahead return conditional on a market distress episode) is given by $\hat{\boldsymbol{\mu}}_{t|\mathrm{SE}} = (\hat{\mu}_{1,t|\mathrm{SE}}, ..., \hat{\mu}_{N,t|\mathrm{SE}})^T$. Similarly, the conditional expected market return $\mu_{m,t|\mathrm{SE}}$ can be estimated as

$$\hat{\mu}_{m,t|\mathrm{SE}} = \frac{\sum_{s=1}^{S} R_{m,t+1}^{(s)}\, I\{R_{m,t+1}^{(s)} < C\}}{\#\mathrm{SE}}. \tag{24}$$

Thus, the estimator of CoER can be written as

$$\mathrm{Co\hat{E}R}_t = \boldsymbol{W}_t^T \hat{\boldsymbol{\mu}}_{t|\mathrm{SE}} - \hat{\mu}_{m,t|\mathrm{SE}}, \tag{25}$$

where $\boldsymbol{W}_t$ denotes the vector of portfolio weights that is known at month t. As for the CoSD, we first estimate the conditional covariance matrix of the vector of asset returns $\mathbf{\Sigma}_{t|\mathrm{SE}}$ using the Monte Carlo sample counterpart, with element (i, j) defined as

$$\hat{\Sigma}_{t(i,j)|\mathrm{SE}} = \frac{\sum_{s=1}^{S} \big(R_{i,t+1}^{(s)} - \hat{\mu}_{i,t}\big)\big(R_{j,t+1}^{(s)} - \hat{\mu}_{j,t}\big) I\{R_{m,t+1}^{(s)} < C\}}{\#\mathrm{SE} - 1}, \tag{26}$$

for $i, j = 1, \ldots, N$. We then estimate the conditional variance of market return $\sigma^2_{m,t|\mathrm{SE}}$ as

$$\hat{\sigma}^2_{m,t|\mathrm{SE}} = \frac{\sum_{s=1}^{S} \big(R_{m,t+1}^{(s)} - \hat{\mu}_{m,t}\big)^2 I\{R_{m,t+1}^{(s)} < C\}}{\#\mathrm{SE} - 1}. \tag{27}$$

Analogously, for each asset i, an estimator of the conditional covariance between asset's i and market returns $\sigma_{im,t|\mathrm{SE}}$ is given by

$$\hat{\sigma}_{im,t|\mathrm{SE}} = \frac{\sum_{s=1}^{S} \big(R_{i,t+1}^{(s)} - \hat{\mu}_{i,t}\big)\big(R_{m,t+1}^{(s)} - \hat{\mu}_{m,t}\big) I\{R_{m,t+1}^{(s)} < C\}}{\#\mathrm{SE} - 1}, \tag{28}$$

thus the estimator of the vector of conditional covariances between individual assets and the market portfolio is $\hat{\boldsymbol{\sigma}}_{t|\mathrm{SE}} = (\hat{\sigma}_{1m,t}, \ldots, \hat{\sigma}_{Nm,t})^T$. Combining the above estimators, we obtain the following estimator of CoSD at month t:

$$\widehat{\mathrm{CoSD}}_t = \Big(\boldsymbol{W}_t^T \hat{\mathbf{\Sigma}}_{t|\mathrm{SE}} \boldsymbol{W}_t + \hat{\sigma}^2_{m,t|\mathrm{SE}} - 2\boldsymbol{W}_t^T \hat{\boldsymbol{\sigma}}_{t|\mathrm{SE}}\Big)^{1/2}. \tag{29}$$

4 Empirical analysis

4.1 Data

Our empirical analysis is conducted using monthly cross-sectional US market data spanning from January 1985 to December 2021, for a period of 37 years. In this section, we first provide details of the predictor set and then discuss the choice of portfolio assets.

4.1.1 Description of predictors

We adopt 94 monthly firm characteristics used by Gu et al. (2020).[3] We manually matched this dataset with CRSP monthly returns. The equities presented in this original dataset compose of listed firms within NASDAQ, AMEX, and NYSE ranging from 1965 to 2021. The detailed specifications of variables can be found in Online Appendix F of Gu et al. (2020).

In addition to stock-level characteristics, we also consider 14 macroeconomic variables. Among those, eight are used in Gu et al. (2020), including dividend-price ratio (macro_dp), earnings-price ratio (macro_ep), book-to-market ratio (macro_bm), net equity expansion (macro_ntis), Treasury-bill rate (macro_tbl), term spread (macro_tms), default spread (macro_dfy), and stock variance (macro_svar); and six are

[3] We computed the value-weighted average of characteristics for the market portfolio using the 500 highest market cap firms. The correlation coefficient between S&P 500 return and our constructed return is beyond 0.99.

uncertainty indices proposed by Ludvigson et al. (2021), which covers total real uncertainty index (macro_TRU), economic real uncertainty index (macro_ERU), total macro uncertainty index (macro_TMU), economic macro uncertainty index (macro_EMU), total financial uncertainty index (macro_TFU), and economic financial uncertainty index (macro_EFU).

Lastly, we consider industry dummies based on the first two digits of the SIC code. To summarize, our predictor set covers 94 firm characteristics, 14 macroeconomic variables, and 74 industry dummies. Throughout our empirical studies, the predictors are defined as follows:

$$z_{i,t} = x_t \otimes c_{i,t}, \tag{30}$$

where $c_{i,t}$ denotes the vector of 94 characteristics for firm i, and x_t represents the vector of macroeconomic variables with an added constant C. Thus, $z_{i,t}$ is the vector of predictors including interactions between macroeconomic variables and stock-level signals. The total number of covariates is $94 \times (14 + 1) + 74 = 1484$.

The original dataset used by Gu et al. (2020) spans from March 1957 to December 2016, covering 60 years of history. However, it includes a large number of missing variables.[4] After deleting the missing data, we obtain a dataset that starts in January 1985 and ends in December 2021, containing 256,632 monthly observations with 577 firms in total.

4.1.2 The choice of portfolio assets

As argued by Lin et al. (2023), big financial institutions are preferred in systemic risk-based portfolio analysis since they are more exposed to market distress than non-financial counterparts. Their pre-analysis results have shown that the objective function of CoSR is more relevant when the universe of portfolio assets covers large financial institutions that are potentially systemic, although not necessarily classified as Systemically Important Financial Institutions (SIFIs). Therefore, in this paper, we consider a set of portfolio assets that includes only large financial institutions.

In November of 2021, the Financial Stability Board (FSB), in consultation with Basel Committee on Banking Supervision and national authorities, identified a list of Global SIFIs (G-SIFIs). The total number of G-SIFIs contained in the FSB's list is 30, among which 5 are traded on the US market throughout our sample period. Besides, the Board of Governors of the US Federal Reserve System maintains a list of Domestic SIFIs (D-SIFIs). This list contains financial firms not big enough to be classified as G-SIFIs, but are still considered to be domestic systemically important. According to the list released by the Federal Reserve as of March 2014, 23 banks traded on the US stock market were identified as D-SIFIs. Among those D-SIFIs, 12 are traded throughout our sample period. Finally, we obtain a list of 17 SIFIs consisting of 5 G-SIFIs and 12 D-SIFIs.

Following Brownlees and Engle (2016), we select large financial firms with a market capitalization greater than 5 bln USD as of the end of June 2007. After applying

[4] All data before January 1985 contains at least one variable with a large portion of missing observations. Thus, it is not possible to fill in those missing values with monthly cross-sectional medians as in Gu et al. (2020). Filling with medians does not seem appropriate since it will negatively affect the explanatory power of some predictors. We, therefore, choose to only consider the sample period without missing observations.

this filter criterion to our dataset, we are left with a list of 38 assets that covers the aforementioned 17 SIFIs. Therefore, we finally obtain a list of 38 portfolio assets including 17 SIFIs and 21 non-SIFIs. These firms are listed in Table 2.

4.2 Estimation and selection of SPNN model

4.2.1 Sample splitting

We predict conditional return quantiles via a recursive procedure. To achieve this, we first divide our original sample into two disjoint but consecutive subsamples. The first subsample - known as in-sample - is further divided into a training subsample $\mathcal{L}_1$ and a validation subsample $\mathcal{L}_2$ that we use to estimate and select the best SPNN model, respectively. The second subsample - known as out-of-sample - represents a testing subsample $\mathcal{L}_3$ on which we make final forecasts. The initial size of our recursive window is set to 180 monthly observations (from January 1985 to December 1999). The increment size of the window is one month, which results in an out-of-sample with 264 monthly observations starting from January 2000 and ending in December 2021.

It is well known that the ML models are prone to overfit the data, so it is crucial to go through a rigorous procedure of model selection. Hyperparameter tuning helps control model complexity and determine model predictive power as well. Following Gu et al. (2020), we use the validation subsample $\mathcal{L}_2$ to perform model selection. Specifically, for each iteration, we use as a validation subsample $\mathcal{L}_2$ the last 20% of cross-sectional data of each in-sample for all 577 firms and the market, with the first 80% of the observations as the training subsample. We estimate the neural network model multiple times using different sets of hyperparameters on $\mathcal{L}_1$. The subsequent $\mathcal{L}_2$ is then employed to determine the optimal tuning parameters by evaluating the quantile forecasts based on the model estimates obtained over $\mathcal{L}_1$ for the respective hyperparameter set. In particular, the quantile score (QS) is adopted for evaluating quantile forecasts, which takes into account both sharpness and reliability; see Hong et al. (2016). It is defined as the mean of pinball losses throughout the forecasting horizon and across all targeted quantile levels:

$$\text{QS} = \frac{1}{\#M \times H \times (N+1)} \sum_{m \in M} \sum_{t=1}^{H} \sum_{j=1}^{N+1} \rho\big(R_{j,t}, \hat{q}_{j,t}(\tau_m)\big), \tag{31}$$

where M is the quantile set of interest (we set $M = \{1, 2, ..., 99\}$), $R_{j,t}$ is the realized return of individual firm or market, and H indicates the forecast horizon ($H = 12$ in our case).

After selecting the best set of hyperparameters, we re-estimate our model using the in-sample data on $\mathcal{L}_1 + \mathcal{L}_2$, based on which we obtain the final quantile forecasts of returns over the out-of-sample period $\mathcal{L}_3$. As for the data preprocessing, we standardize features by removing the mean and scaling to unit variance. The data is first normalized within each of the training subsamples during hyperparameter tuning and then normalized for observations within the whole in-sample when making final forecasts. Due to the computational intensity of ML-based approaches, we re-fit our model once a year and retain the corresponding estimates to obtain the quantile forecasts for that year; see Gu et al. (2020) and Kynigakis and Panopoulou (2021).

4.2.2 SPNN configuration

We consider neural networks with up to three hidden layers. In particular, we consider the following specifications: (1) SPNN model that has a single hidden layer with 32 neurons (hereafter SPNN1); (2) SPNN model that has two hidden layers with 32 and 16 neurons (hereafter SPNN2); and (3) SPNN model that has three hidden layers with 32, 16, and 8 neurons (hereafter SPNN3).

In practice, we adopt the Rectified Linear Unit (ReLU) $g(x) = \max(0, x)$ as the activation function of hidden layers, which promotes sparsity in the number of active neurons and allows for an efficient derivative computation as well; see Nair and Hinton (2010) among others. For the output layer, we apply the identity activation function $g(x) = x$ following Hatalis et al. (2019). During model training, we follow Gu et al. (2020) and adopt the Adam optimizer, which is an extended version of the gradient descent method.

4.2.3 Training and regularization methods

The network training is time-consuming due to the high degree of computational complexity involved in tuning abundant parameters and processing mass data. To improve the generalization power of fitted SPNN models and reduce the training cost, in addition to applying l_1 penalization, we consider additional DL techniques including batch training, batch normalization, and early stopping. Finally, to reduce the prediction variance incurred during the stochastic optimization process, we adopt an ensemble approach to initialize neural network models using different random seeds and average forecasts over all models.

4.2.4 Hyperparameters

We use a two-dimensional grid search approach to find the optimal set of hyperparameters by minimizing the QS among all possible SPNN configurations over the validation set $\mathcal{L}_2$. The tuning parameters are the L_1 penalty parameter λ and the learning rate of Adam optimizer lr. For the grid of values we keep following Gu et al. (2020) and set $\lambda \in [10^{-5}, 10^{-3}]$ and $lr \in [10^{-3}, 10^{-2}]$.

Our goal of model selection is modest in the sense of fixing various hyperparameters in advance, though tuning on a larger set of hyperparameters might help in improving accuracy.[5] Unlike Gu et al. (2020) who set the batch size to 10,000, we adopt a relatively small batch size of 32. Although a large batch tends to give more precise estimates of the gradients, a small batch ensures that each training iteration is fast and reduces memory usage as well. Keskar et al. (2016) argued that using a large batch tends to suffer from a generalization drop due to sharp minima, see also Masters and Luschi (2018) and others for the preference for a small batch. For the remaining hyperparameters, we just follow Gu et al. (2020). Specifically, the number of maximum

[5] We also tested for different combinations of l_1-penalty, learning rate, dropout rate, and patience in early stopping, and the current setting is found to be most effective.

epochs is set to 100, the patience in early stopping is set to 5, and the number of ensemble models is set to 10.

4.3 Portfolio formation

After fitting SPNN models, we obtain quantile forecasts of monthly returns, based on which we estimate the conditional marginal return distributions following the method discussed in Section 3.2.1. Combining the distributional forecasts with the fitted t-copula model, we generate 30,000 return scenarios at the beginning of each month over the out-of-sample period.

The optimization problem (17) is solved on a monthly basis by maximizing CoSR measure. Specifically, we estimate the reward and risk measures by computing the first and second conditional moments based on the filtered realizations that satisfy SE conditions. Following Acharya et al. (2017) and Brownlees and Engle (2016), we choose two different SE thresholds C: i) $C = VaR^m_{5\%}$ indicating the most that the financial market loses with 95% confidence over the next month, and ii) $C = -6.7\%$, which corresponds to a 40% decrease in the market index over a six-month period.

For the comparison purpose, we assess the out-of-sample performance of our approach against three benchmark portfolios, namely sample-based SR portfolio, sample-based GMV, and 1/N portfolio.[6] In addition, we also consider S&P 500 Index as a fundamental benchmark. We assume that our investors have an initial wealth of $\mathrm{FW}_0 = 1$ and an initial cumulative log return $\mathrm{CR}_0 = 0$ at the beginning of the backtesting period (December 1999).

Three main steps are performed to calculate the ex-post final wealth and cumulative return at the k-th recalibration ($k = 0, ..., 263$). Firstly, we generate return scenarios based on the algorithms described in Section 3.2.2, and obtain the solution $\boldsymbol{W}^*_{k+1}$ to the optimization problem in (13) for each of the performance measures under consideration. This step is performed using the Matlab built-in function *fmincon*. Following Kresta et al. (2015), we randomly choose 20 starting points in order to approach the global optimum when solving (13). Secondly, the ex-post final wealth is calculated as

$$\mathrm{FW}_{k+1} = \mathrm{FW}_k(1 + \boldsymbol{W}^{*T}_k \boldsymbol{R}_{k+1}), \tag{32}$$

where $\boldsymbol{R}_{k+1}$ is the ex-post vector of simple returns between k and $k+1$. Thirdly, the ex-post cumulative log return is calculated as

$$\mathrm{CR}_{k+1} = \mathrm{CR}_k + \ln(1 + \boldsymbol{W}^{*T}_k \boldsymbol{R}_{k+1}). \tag{33}$$

Note that the latter equation reports the cumulative performance of the portfolio net of wealth. That is, expression (32) implies that $\mathrm{FW}_{K+1} = \mathrm{FW}_0 \prod_{k=0}^{K} (1 + \boldsymbol{W}^{*T}_k \boldsymbol{R}_{k+1})$. Taking logs from the left and right-hand sides of the latter equation, we obtain $(\ln \mathrm{FW}_{K+1} - \ln \mathrm{FW}_0) = \sum_{k=0}^{K} \ln(1 + \boldsymbol{W}^{*T}_k \boldsymbol{R}_{k+1})$. Therefore, the growth in wealth due to the cumulative return on the portfolio is given by expression (33). By repeatedly

[6] To reduce the estimation error of sample covariance matrix, we applied the shrinkage estimator proposed by Ledoit and Wolf (2004) to SR portfolio and GMV.

computing FW_{k+1} and CR_{k+1}, we obtain the wealth and cumulative return paths over the backtesting period.

4.4 Results

In this section, we first illustrate return quantile forecasts and then examine the predictive power of covariates using two variable importance measures. After that, we report backtesting results with and without transaction costs. Finally, we compare the level of systemic risk of different strategies under investigation.

4.4.1 Quantile forecasts and variable importance

To present some insights on the return quantile forecasts obtained using SPNN models, in Figure 3 we display the realized returns and the prediction intervals obtained using SPNN1. To further save space, we only show results for the S&P 500 Index below.[7] From Figure 3, we see that the return quantile forecasts are able to capture most of the variation in the realized returns over the out-of-sample period, especially during crisis episodes.

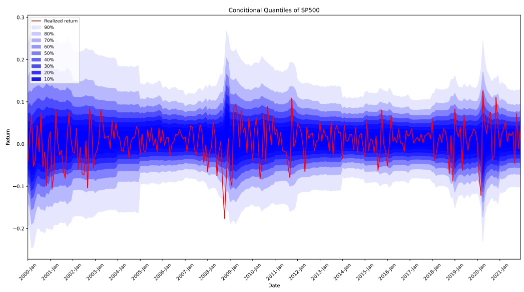


Fig. 3: Conditional quantiles of market returns by SPNN1.

Next, we investigate the relative importance of individual predictors for the performance of SPNN model on both training and testing sets. Gu et al. (2020) highlighted the importance of ranking the variable importance of predictors to add interpretability to ML-based models. Unlike Gu et al. (2020) and Kynigakis and Panopoulou (2021) who use the change in the out-of-sample R^2 to measure the variable importance in the context of mean regression, hereafter we adopt two measures that are more suitable for measuring performance related to quantile forecasts.

We first consider the Mean Squared Sensitivity (MSS), which measures the sensitivity of the output of the m-th neuron in the output layer with respect to the p-th

[7]The corresponding results for portfolio assets are available upon request.

input predictor (Zurada et al. 1994; Yeh and Cheng 2010):

$$\text{MSS}_{p,m} = \sqrt{\frac{\sum_{t\in(\mathcal{L}_1+\mathcal{L}_2)}\big(s_{p,m}|_{\boldsymbol{X}_t}\big)^2}{|\mathcal{L}_1|+|\mathcal{L}_2|}}, \tag{34}$$

with

$$s_{p,m}\big|_{\boldsymbol{X}_t} = \frac{\partial \hat{Q}_{R_{t+1}}(\tau_m|\boldsymbol{X}_t)}{\partial x_{p,t}}(\boldsymbol{X}_t), \tag{35}$$

where $\boldsymbol{X}_t = (x_{1,t}, ..., x_{P,t})^T$ refers to the t-th observation of the P predictors in the in-sample $(\mathcal{L}_1 + \mathcal{L}_2)$ on which we perform the sensitivity analysis, $s_{p,m}\big|_{\boldsymbol{X}_t}$ refers to the sensitivity of the output of the m-th neuron in the output layer (which in our case is the τ_m-th conditional quantile) with respect to the input of the p-th neuron in the input layer evaluated at $\boldsymbol{X}_t$, and $|\mathcal{L}_i|$ denote the number of observations in set $\mathcal{L}_i$, for $i = \{1, 2\}$. The sensitivities defined in (35) can be calculated using the chain rule for the partial derivatives of the inner layers, see Pizarroso et al. (2020) for more computational details. By computing MSS, we can measure the sensitivity of model estimation/prediction to the changes in a candidate predictor. In practice, for each predictor x_p, we compute the following average MSS

$$\tilde{\text{MSS}}_p = \frac{1}{M}\sum_{m=1}^{M} \text{MMS}_{p,m}, \tag{36}$$

which allows us to measure the variable importance across all quantiles of interest.

Next, we consider the QRNN causality measure developed by Lin and Taamouti (2023), which is an extension of the Quantile Causality (QC) measure proposed by Song and Taamouti (2021). Specifically, for $\tau \in (0, 1)$, the QC of the p-th input variable in QRNN is defined as

$$\text{QC}_p(\tau) = \ln\left[\frac{E\big[\rho_\tau\big(R_{t+1} - Q_{R_{t+1}}(\tau|\overline{\boldsymbol{X}}_t)\big)\big]}{E\big[\rho_\tau\big(R_{t+1} - Q_{R_{t+1}}(\tau|\boldsymbol{X}_t)\big)\big]}\right], \tag{37}$$

where $\overline{\boldsymbol{X}}_t$ denotes the information set at time t on all predictors, except the p-th predictor. $\text{QC}_p(\tau)$ measures the degree of causal effect from a certain predictor p to the τ-th quantile of the predictand given the past of the latter. As pointed out by Song and Taamouti (2021), QC can be viewed as a measure of the amount of information brought by the past of the p-th predictor to improve the prediction of the τ-th quantile of asset return R_{t+1}. Similar to the average measure $\tilde{\text{MSS}}_p$, in our empirical analysis we compute the average QC for each predictor x_p as

$$\tilde{\text{QC}}_p = \ln\left[\frac{\frac{1}{M|\mathcal{L}_3|}\sum_{m=1}^{M}\sum_{t\in\mathcal{L}_3}\rho_{\tau_m}\big(R_{t+1} - \hat{Q}_{R_{t+1}}(\tau_m|\overline{\boldsymbol{X}}_t)\big)}{\frac{1}{M|\mathcal{L}_3|}\sum_{m=1}^{M}\sum_{t\in\mathcal{L}_3}\rho_{\tau_m}\big(R_{t+1} - \hat{Q}_{R_{t+1}}(\tau_m|\boldsymbol{X}_t)\big)}\right], \tag{38}$$

where the marginal contribution of each predictor x_p is assessed using the out-of-sample $\mathcal{L}_3$ only, whose data does not overlap with those of training or tuning samples.

Figure 4 reports the variable importance measured by MSS for the 10 most influential firm-level predictors and for all macroeconomic variables based on SPNN1 model, while Figure 7 in Appendix B reports the corresponding variable importance results measured by QC.[8] The variable importance is normalized to sum up to one, which makes it easier to interpret the relative importance of the predictive power of each predictor compared to those of others. Variables are ranked such that those with the highest importance are at the top and the lowest are at the bottom.

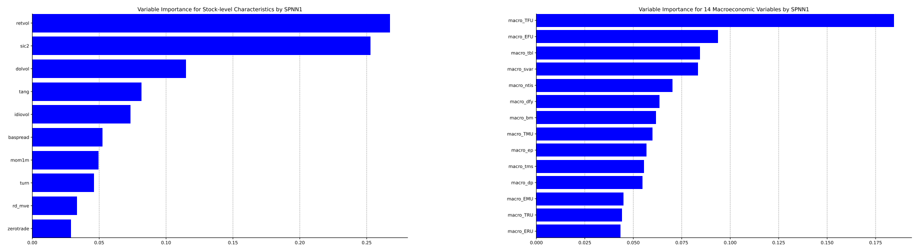


Fig. 4: Left panel displays the top-10 most influential firm-level predictors in SPNN1 measured by MSS, while the right panel reports the corresponding results for all macroeconomic variables.

The top-10 most influential firm-level features measured by MSS as displayed in the top panel of Figure 4 can be grouped into five categories. The first group contains risk measures including the total and idiosyncratic return volatilities (retvol, idiovol); Next are liquidity variables including dollar volume (dolvol), debt capacity/firm tangibility (tang), bid-ask spread (baspread), turnover (turn), and the number of zero trading days (zerotrade); A single momentum predictor constitutes the third group, which is the short-term reversal (mom1m); The fourth group includes a valuation ratio, which is the R&D expense-to-market ratio; The last group consists of industry dummy (sic2). As for the corresponding results of macroeconomic variables, we see from the bottom panel of Figure 4 that all contribute significantly to model training. Among those, the total financial uncertainty index (macro_TFU) is identified as the most influential macro-level predictor.

Analogously, the rankings based on QC measure as shown in Figure 7 draw a similar conclusion. The results agree on a relatively small set of dominant firm-level predictors, which covers the risk measures of total and idiosyncratic return volatilities (retvol, idovol), the liquidity variables of dollar volume (dolvol), industry-adjusted size (mve_ia), bid-ask spread (baspread) and turnover (turn), the short-term reversal (mom1m), and an accounting variable that indicates the number of years since first Compustat coverage (age). While for the macro state variables, the results again

[8] To save space, hereafter we only report the variable importance results obtained by the SPNN1 model. The corresponding results for other SPNN configurations are similar and are available upon request.

confirm their predictive power and place the greatest emphasis on the total financial uncertainty index (macro_TFU) in both cases.

To better illustrate the variable importance over recursive windows, we display the time-varying rankings of predictors in SPNN1 measured by MSS and QC in Figures 8 - 10, consecutively. In particular, we rank the importance of individual predictors according to their average contribution over all quantiles of the returns and across all recursive in-sample or out-of-sample windows depending on the measure of use. Columns in these figures correspond to the start year in each window, and the color gradient within each column indicates the most influential (dark blue) to least influential (light blue) predictors.

4.4.2 Backtesting results

In this section, we conduct a backtesting analysis to assess the economic gains of using our portfolio selection approach. To do so, we compare the out-of-sample performance of SPNN-CoSR portfolios with those of benchmark portfolios under consideration. The optimized portfolios were built recursively using conditional/unconditional return moments estimated from simulated/historical return observations at each iteration starting in January 2000. All portfolios are monthly rebalanced until the end of the out-of-sample period (i.e. December 2021).

The backtesting results are displayed in Figure 5. There are several noticeable features. First, all candidate portfolios outperform the market index. Second, all portfolios did poorly during the 2007-2008 financial crisis. SR, GMV and 1/N strategies lose almost all of their values during that period, while SPNN-CoSR portfolios perform significantly better than others, even though they lost around half of their values since the last peak in 2007. In particular, among SPNN-CoSR portfolios, the SPNN1-based ones deliver the best out-of-sample performance. Third, all SPNN-based CoSR portfolios show a strong upward trend in profitability throughout the evaluation period, which can be mainly attributed to their resilience during market distress. The backtesting results confirm the benefits of combining SPNN-based return forecasts with the incorporation of systemic risk into traditional mean-variance framework when constructing optimal portfolios.

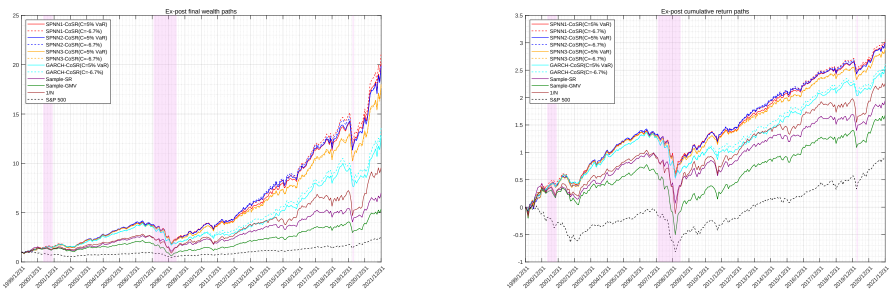


Fig. 5: Paths of final wealth (left panel) and cumulative return (right panel).

Table 3 reports the values of several performance metrics. The results vary among different strategies depending on the choice of performance measure, with the exception is 1/N portfolio which does not rely on any optimization problem or model estimation. The SPNN-CoSR portfolios dominate all other benchmark strategies in terms of profitability, whichever model configuration is being considered. Among those, the SPNN1-based CoSR portfolio with $C = -6.7\%$ delivers the highest value by the end of the evaluation period. Besides, the GARCH-based CoSR portfolios are serious competitors which provide comparable profits over the out-of-sample period but still cannot beat our proposed approach. Specifically, our investors would multiply their wealth by 20.145 and 21.216 using SPNN1-based CoSR portfolios with $C = VaR^m_{5\%}$ and $C = -6.7\%$, respectively, which is almost twice that of GARCH-based CoSR portfolios with $C = VaR^m_{5\%}$ (12.797) and $C = -6.7\%$ (13.650). The GMV gives the lowest final wealth (5.348) and annual return (0.079), while the sample-based SR portfolio performs as the second-worst with a final wealth of 6.973 and an annual return of 0.092. Interestingly, the naive 1/N strategy outperforms all sample-based portfolios in terms of profitability, with the former exhibiting a final wealth of 9.541 and an annual return of 0.108.

The ex-post Sharpe ratio, Sortino ratio and Calmar ratio again demonstrate the superiority of our proposed approach. In particular, the SPNN1-based CoSR portfolio with $C = -6.7\%$ delivers the highest values for all performance ratios among candidate portfolios. In addition, we also test the significance of the difference in Sharpe ratios between SPNN1-based CoSR portfolio and that of each benchmark strategy following Ledoit and Wolf (2008). The relevant results are reported in Table 1. The testing results further confirm the enhanced portfolio performance of our approach.

Table 1: Testing statistics

	SPNN1-based CoSR (C=$VaR^m_{5\%}$)			SPNN1-based CoSR (C=-6.7%)		
Strategy	p value	$\hat{\Delta}$	Original	p value	$\hat{\Delta}$	Original
Sample-based SR	0.005	0.104	3.285	0.005	0.108	3.727
Sample-based GMV	0.000	0.126	4.468	0.000	0.130	5.144
1/N	0.017	0.100	3.120	0.009	0.104	3.336

Besides the above-mentioned performance ratios, investors may consider alternative statistics to gain deeper insights into their trading strategies. Therefore, we add Maximum Drawdown (MDD), Maximum One-Month loss (MOL), and average Turnover (TO) as additional performance metrics. Formally, the MDD is defined as

$$\text{MDD} = \max_{t_0 \le t_1 \le t_2 \le T_0} \left\{ r_{p,t_0:t_1} - r_{p,t_0:t_2} \right\}, \tag{39}$$

where $r_{p,t_0:t}$ denotes the cumulative portfolio return from time t_0 to t_i, for $i \in \{1, 2\}$, with t_0 and T_0 being the first and last months of evaluation period. MOL measures the largest decline in portfolio value over a one-month period, and the average TO is defined as

$$\mathrm{TO} = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}\left|\omega_{i,t+1} - \frac{\omega_{i,t}(1+R_{i,t+1})}{1+\sum_{j=1}^{N}\omega_{j,t}R_{j,t+1}}\right|\right), \tag{40}$$

where $\omega_{i,t}$ is the desired weight of portfolio asset i at time t.

Table 3 also reports the values of these alternative measures. While the SPNN1-based CoSR portfolio with $C = -6.7\%$ provides the highest profitability, it has the lowest MDD as well. Furthermore, all SPNN-based CoSR portfolios regardless of model configurations and SE thresholds outperform other competitors in terms of MDD, which demonstrates the better performance of our proposed approach during market distress. Next, the values of MOL of SPNN-based CoSR portfolios are lower than those of GARCH-based counterparts, while the sample-based GMV displays the lowest MOL since it focuses on the risk only.

4.4.3 Effects of transaction costs

The estimation of transaction cost (TC) is based on TO as defined in (40). After accounting for a proportional TC of c, the portfolio return is now calculated as follows:

$$\tilde{R}_{p,t+1} = (1+R_{p,t+1})\left(1 - c\sum_{i=1}^{N}\left|\omega_{i,t+1} - \frac{\omega_{i,t}(1+R_{i,t+1})}{1+\sum_{j=1}^{N}\omega_{j,t}R_{j,t+1}}\right|\right) - 1. \tag{41}$$

Gu et al. (2020) argued that, given the large role of price trend predictors employed by ML-based approaches, it is unsurprising that the ML-based trading strategies are characterized by relatively high TO. This also holds for our SPNN-based approach as shown in Table 3. Although our SPNN-based CoSR portfolios show a higher TO than that of the sample-based benchmarks, their values are still much lower than those of the GARCH-based CoSR portfolios. Unsurprisingly, the 1/N portfolio delivers the lowest TO due to its well-diversified property.

Although the CoSR portfolios with relatively high TO are more flexible to adapt to the changes in market conditions than other benchmarks, their portfolio values are likely to decrease due to their higher TC during rebalancing. To analyze the effects of TC, we set a relatively high value of $c = 50$ basis points (bps) and recompute the ex-post paths of final wealth and cumulative return for all portfolios under consideration. Table 4 reports the values of the performance measures after taking into account proportional TC.[9] In short, the inclusion of TC does not change our main conclusions. The SPNN-based CoSR portfolios still outperform all other competitors in terms of profitability. Remarkably, the final wealth of SPNN-based CoSR portfolios is still more than twice that of other benchmarks excluding the 1/N portfolio.

[9] To save space, the Figure 11 that illustrates the backtesting results after considering transaction costs was moved to the Appendix B.

4.4.4 Portfolio-level systemic risk

In this section, we measure the portfolio-level systemic risk using portfolio's LRMES proposed by Lin et al. (2023), which is defined as

$$\mathrm{LRMES}_{p,t} = \sum_{i=1}^{N} \omega_{i,t}\, \mathrm{LRMES}_{i,t}, \tag{42}$$

where $\mathrm{LRMES}_{i,t}$ indicates the expected loss in equity value of asset i over month t. The portfolio's LRMES can be interpreted as the expected percentage drop in portfolio value under stressed market conditions. Figure 6 illustrates the portfolio-level LRMES over the evaluation period.[10] Overall speaking, the SPNN-based CoSR portfolios give the best performance in terms of systemic risk measured by LRMES. The relatively low portfolio-level LRMES indicates less potential loss during crisis periods. Specifically, the SPNN-based CoSR portfolio with $C = -6.7\%$ presents the lowest LRMES than all other competitors throughout the out-of-sample period, while all benchmark strategies provide much higher and volatile LRMES values.

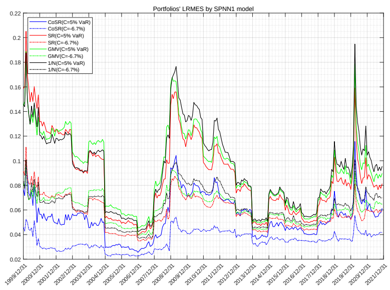


Fig. 6: Portfolio-level LRMES by SPNN1.

5 Conclusion

We explore whether using return forecasts generated via smooth pinball quantile regression neural network can add value to systemic risk-based portfolio selection. The optimal portfolio is constructed by maximizing an ex-ante conditional Sharpe ratio based on simulated return scenarios, and its out-of-sample performance is compared with that of tangency portfolio, minimum variance portfolio, and equally-weighted portfolio. The proposed approach outperforms all other benchmarks in terms of profitability and portfolio-level systemic risk. The testing results of the difference of Sharpe ratios further confirm its significant outperformance against benchmark strategies. Although our portfolio is characterized by a relatively high turnover rate, its superiority is still tenable after accounting for a considerable amount of proportional

[10]To save space, we only show the results obtained by SPNN1 here. However, the corresponding results for other SPNN configurations are available upon request.

transaction costs. Another side contribution of our paper is the implementation of two variable importance measures, which we propose to rank the most influential predictors in SPNN models. The relevant results demonstrate the substantial predictive information brought by macroeconomic variables, whereas only a limited number of firm-level signals contribute to the training and prediction process.

Appendix A

Table 2: Portfolio assets

Firm name	**Ticker**
Synovus Financial Corp.	SNV
Jefferies Financial Group Inc.	JEF
Cincinnati Financial Corporation	CINF
Comerica Incorporated	CMA
Loews Corporation	L
Vornado Realty Trust	VNO
Fifth Third Bancorp	FITB
Regions Financial Corporation	RF
M&T Bank Corporation	MTB
Franklin Resources, Inc.	BEN
Wells Fargo & Company	WFC
Huntington Bancshares Incorporated	HBAN
Marsh & McLennan Companies, Inc.	MMC
Host Hotels & Resorts, Inc.	HST
CNA Financial Corporation	CNA
JPMorgan Chase & Co.	JPM
Humana Inc.	HUM
Lincoln National Corporation	LNC
The Bank of New York Mellon Corporation	BK
Aflac Incorporated	AFL
Northern Trust Corporation	NTRS
American Express Company	AXP
Bank of America Corporation	BAC
The PNC Financial Services Group, Inc.	PNC
Aon plc	AON
Globe Life Inc.	GL
Cigna Corporation	CI
The Progressive Corporation	PGR
Public Storage	PSA
KeyBank	KEY
U.S. Bancorp	USB
SLM Corporation	SLM
American International Group, Inc.	AIG
SEI Investments Company	SEIC
Truist Financial Corporation	TFC
State Street Corporation	STT
Zions Bancorporation	ZION
UnitedHealth Group Incorporated	UNH

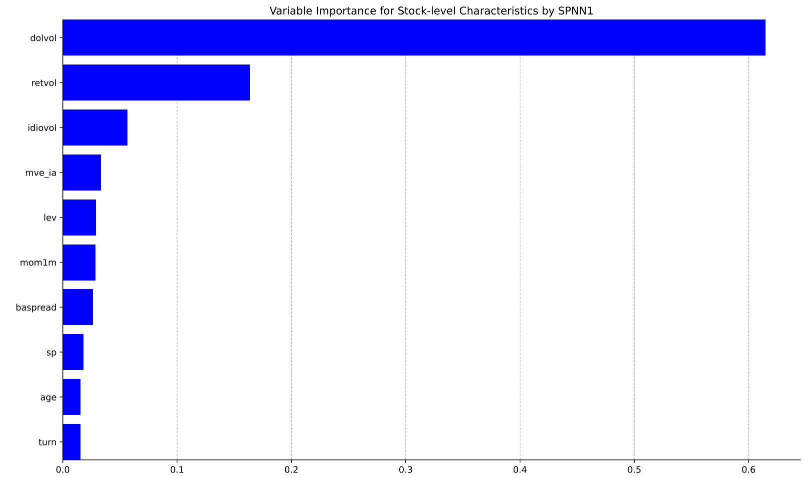
Table 3: Backtesting results without transaction costs

Strategy	Final wealth	Annual return	MDD	MOL	TO	Sharpe ratio	Sortino ratio	Calmar ratio
SPNN1-based CoSR (C=$VaR^m_{5\%}$)	20.145	0.146	0.549	0.203	0.125	0.754	1.237	0.266
SPNN1-based CoSR (C=-6.7%)	21.216	0.149	0.527	0.205	0.110	0.767	1.257	0.283
SPNN2-based CoSR (C=$VaR^m_{5\%}$)	19.908	0.146	0.587	0.205	0.127	0.742	1.199	0.248
SPNN2-based CoSR (C=-6.7%)	19.965	0.146	0.566	0.208	0.110	0.753	1.216	0.258
SPNN3-based CoSR (C=$VaR^m_{5\%}$)	18.002	0.140	0.557	0.202	0.128	0.734	1.204	0.252
SPNN3-based CoSR (C=-6.7%)	17.770	0.140	0.534	0.202	0.115	0.730	1.202	0.262
GARCH-based CoSR (C=$VaR^m_{5\%}$)	12.797	0.123	0.568	0.226	0.392	0.635	1.035	0.216
GARCH-based CoSR (C=-6.7%)	13.650	0.126	0.557	0.252	0.384	0.651	1.055	0.226
Sample-based SR	6.973	0.092	0.602	0.210	0.038	0.394	0.661	0.153
Sample-based GMV	5.348	0.079	0.722	0.196	0.028	0.317	0.546	0.110
1/N	9.541	0.108	0.686	0.245	0.024	0.408	0.667	0.157

Table 4: Backtesting results with 50 bps proportional transaction costs

Strategy	Final wealth	Annual return	MDD	MOL	Sharpe ratio	Sortino ratio	Calmar ratio
SPNN1-based CoSR (C=$VaR^m_{5\%}$)	14.462	0.129	0.562	0.204	0.660	1.081	0.230
SPNN1-based CoSR (C=-6.7%)	15.855	0.134	0.539	0.206	0.685	1.119	0.248
SPNN2-based CoSR (C=$VaR^m_{5\%}$)	14.225	0.128	0.601	0.208	0.646	1.043	0.214
SPNN2-based CoSR (C=-6.7%)	14.912	0.131	0.580	0.211	0.668	1.078	0.225
SPNN3-based CoSR (C=$VaR^m_{5\%}$)	12.834	0.123	0.572	0.205	0.635	1.041	0.215
SPNN3-based CoSR (C=-6.7%)	13.103	0.124	0.548	0.204	0.641	1.055	0.226
GARCH-based CoSR (C=$VaR^m_{5\%}$)	4.534	0.071	0.601	0.230	0.338	0.586	0.118
GARCH-based CoSR (C=-6.7%)	4.941	0.075	0.591	0.256	0.362	0.618	0.127
Sample-based SR	6.309	0.087	0.605	0.210	0.370	0.624	0.144
Sample-based GMV	4.971	0.076	0.725	0.197	0.299	0.519	0.104
1/N	8.950	0.105	0.689	0.246	0.394	0.645	0.152

Appendix B



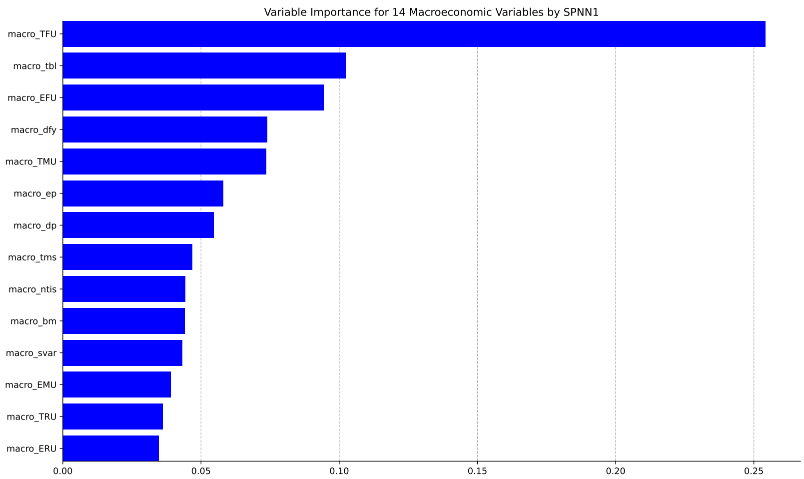
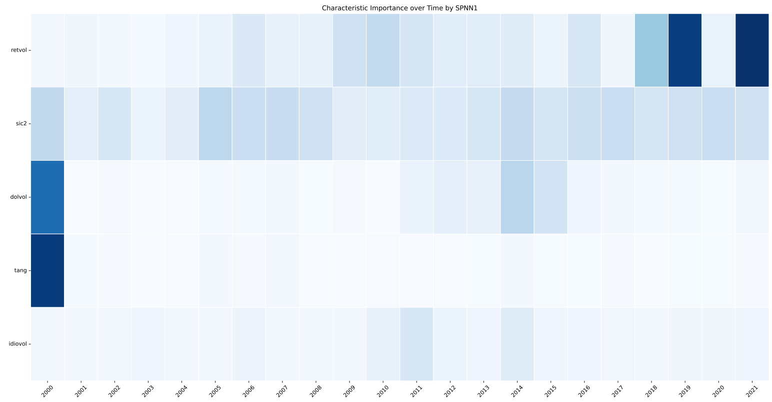


Fig. 7: Top and bottom panels display the variable importance of top-10 most influential firm-level predictors and all macroeconomic variables measured by QC using SPNN1, respectively.



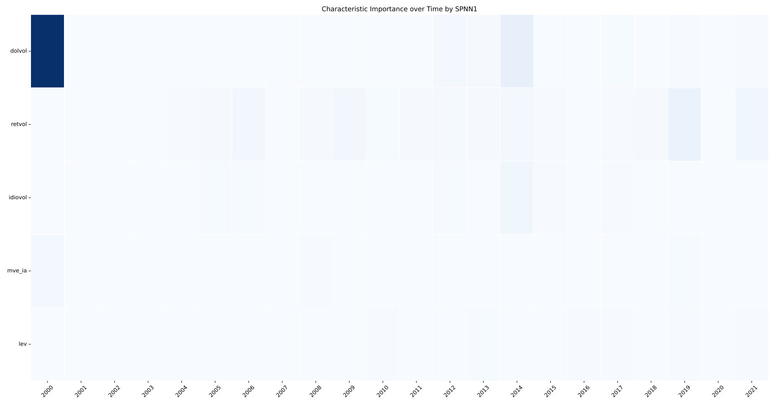


Fig. 8: Time-varying variable importance of the top-5 most influential firm-level predictors measured by MSS (top panel) and QC (bottom panel). Predictors are ordered based on the average value of their MSS over recursive training, with the most influential features at the top and the least influential at the bottom. Columns correspond to the year-end of the 22 in-sample windows, and color gradients within each column indicate the most influential (dark blue) to least influential (white) variables.

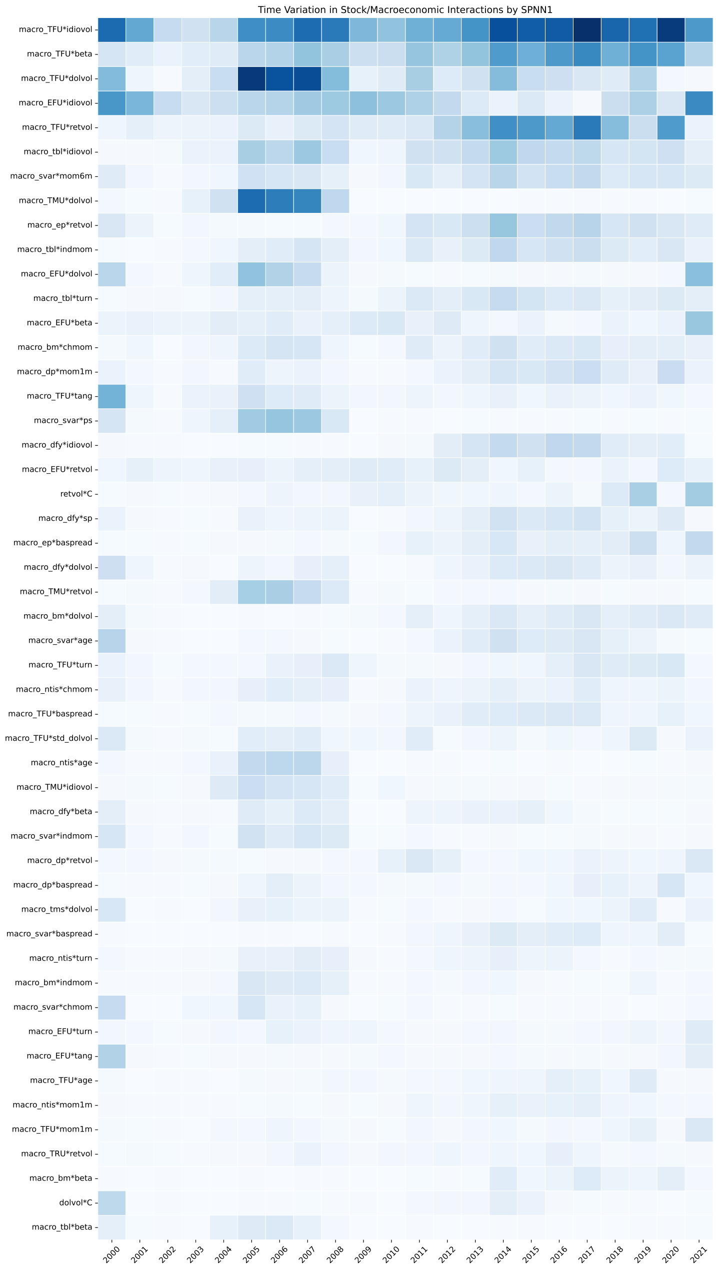


Fig. 9: Time-varying variable importance of the top-50 most influential predictors of interactions between each firm characteristic with macroeconomic variables measured by MSS. Columns correspond to the year-end of the 22 in-sample windows, and color gradients within each column indicate the most influential (dark blue) to least influential (white) variables.

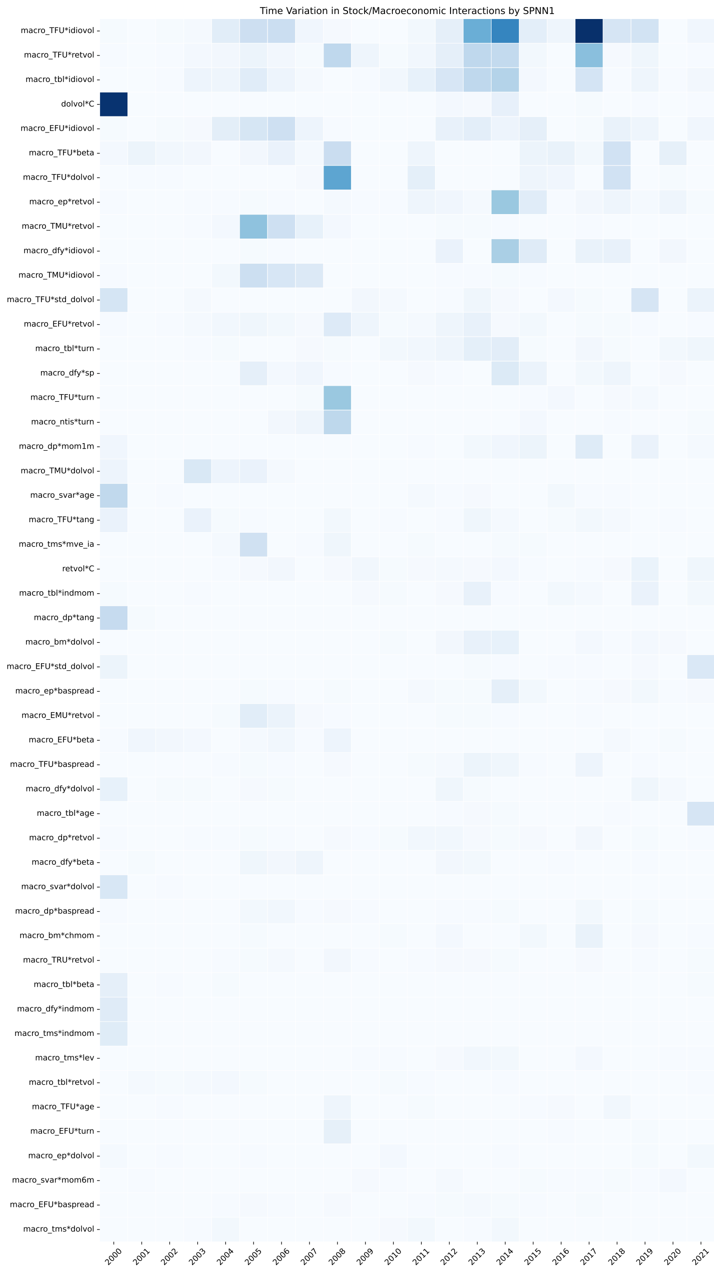
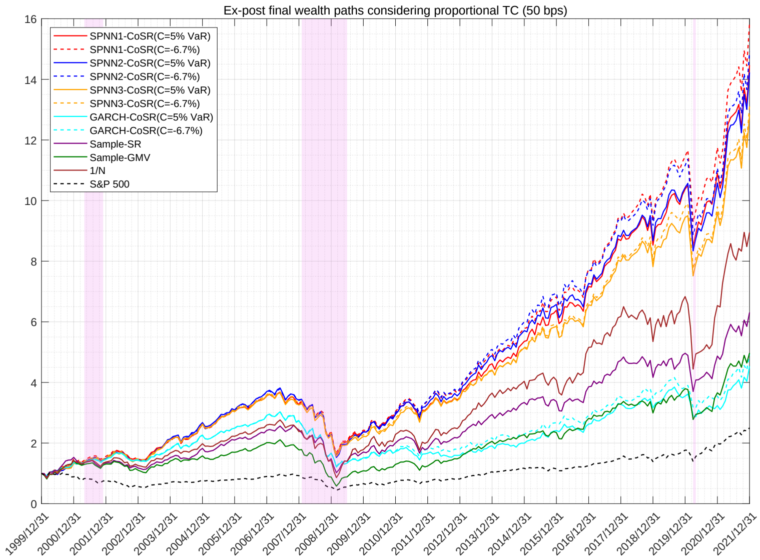


Fig. 10: Time-varying variable importance of the top-50 most influential predictors of interactions between each firm characteristic with macroeconomic variables measured by QC. Columns correspond to the year start of each of the 22 out-of-sample windows, and color gradients within each column indicate the most influential (dark blue) to least influential (white) variables.



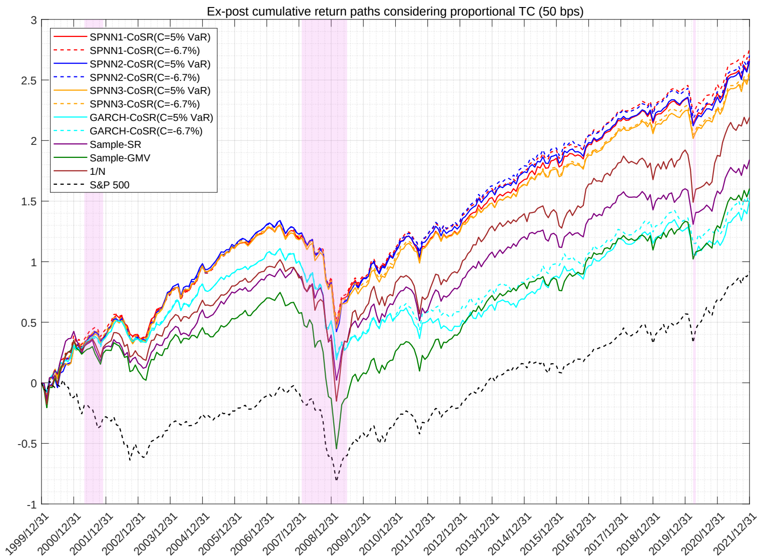


Fig. 11: Ex-post final wealth (top panel) and ex-post cumulative return (bottom panel) paths obtained using different strategies with 50 bps proportional transaction costs. The shaded areas denote recession periods as defined by NBER.

References

Acharya, V.V., L.H. Pedersen, T. Philippon, and M. Richardson. 2017. Measuring systemic risk. *The Review of Financial Studies 30*(1): 2–47 .

Arends, E.L., S.J. Watson, S. Basu, and B. Cheneka 2020. Probabilistic wind power forecasting combining deep learning architectures. In *2020 17th International Conference on the European Energy Market (EEM)*, pp. 1–6. IEEE.

Babiak, M. and J. Baruník. 2020. Deep learning, predictability, and optimal portfolio returns. *arXiv preprint arXiv:2009.03394* .

Bang, S., S.H. Eo, Y.M. Cho, M. Jhun, and H. Cho. 2016. Non-crossing weighted kernel quantile regression with right censored data. *Lifetime Data Analysis 22*(1): 100–121 .

Branger, N., K. Lučivjanská, and A. Weissensteiner. 2019. Optimal granularity for portfolio choice. *Journal of Empirical Finance* 50: 125–146 .

Brownlees, C. and R.F. Engle. 2016. SRISK: A conditional capital shortfall measure of systemic risk. *The Review of Financial Studies 30*(1): 48–79 .

Cannon, A.J. 2011. Quantile regression neural networks: implementation in R and application to precipitation downscaling. *Computers & Geosciences* 37: 1277–1284. doi:10.1016/j.cageo.2010.07.005 .

Cannon, A.J. 2018. Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic Environmental Research and Risk Assessment 32*(11): 3207–3225 .

Capponi, A. and A. Rubtsov. 2022. Systemic risk-driven portfolio selection. *Operations Research* .

DeMiguel, V., L. Garlappi, and R. Uppal. 2009. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies 22*(5): 1915–1953 .

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

Harvey, C.R., Y. Liu, and H. Zhu. 2016. ... and the cross-section of expected returns. *The Review of Financial Studies 29*(1): 5–68 .

Hatalis, K., A.J. Lamadrid, K. Scheinberg, and S. Kishore. 2019. A novel smoothed loss and penalty function for noncrossing composite quantile estimation via deep neural networks. *arXiv preprint arXiv:1909.12122* .

Hong, T., P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R.J. Hyndman. 2016. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting 32*(3): 896–913 .

Huang, X., M. Guidolin, E. Platanakis, and D. Newton. 2021. Dynamic portfolio management with machine learning. *Available at SSRN 3770688* .

Huber, P.J. 2004. *Robust statistics*, Volume 523. John Wiley & Sons.

Keskar, N.S., D. Mudigere, J. Nocedal, M. Smelyanskiy, and P.T.P. Tang. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836* .

Koenker, R. and G. Bassett. 1978. Regression quantiles. *Econometrica: Journal of the Econometric Society*: 33–50 .

Kresta, A. et al. 2015. Application of performance ratios in portfolio optimization. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis 63*(6): 1969–1977 .

Kynigakis, I. and E. Panopoulou. 2021. Does model complexity add value to asset allocation? Evidence from machine learning forecasting models. *Journal of Applied Econometrics* .

Ledoit, O. and M. Wolf. 2004. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management 30*(4): 110–119 .

Ledoit, O. and M. Wolf. 2008. Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance 15*(5): 850–859 .

Lin, W., J. Olmo, and A. Taamouti. 2023. Portfolio selection under systemic risk. *Journal of Money, Credit and Banking (forthcoming)* .

Lin, W. and A. Taamouti 2023. Measuring Granger causality in quantile regression neural network. Technical report, Working paper, Durham University.

Ludvigson, S.C., S. Ma, and S. Ng. 2021. Uncertainty and business cycles: exogenous impulse or endogenous response? *American Economic Journal: Macroeconomics 13*(4): 369–410 .

Markowitz, H. 1952. Portfolio selection. *The Journal of Finance 7*(1): 77–91 .

Masters, D. and C. Luschi. 2018. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612* .

McNeil, A.J., R. Frey, and P. Embrechts. 2015. *Quantitative risk management: concepts, techniques and tools-revised edition.* Princeton university press.

Nair, V. and G.E. Hinton 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814.

Ouali, D., F. Chebana, and T.B. Ouarda. 2016. Quantile regression in regional frequency analysis: a better exploitation of the available information. *Journal of Hydrometeorology 17*(6): 1869–1883 .

Pizarroso, J., J. Portela, and A. Muñoz. 2020. NeuralSens: sensitivity analysis of neural networks. *arXiv preprint arXiv:2002.11423* .

Rachev, S., S. Ortobelli, S. Stoyanov, F.J. Fabozzi, and A. Biglova. 2008. Desirable properties of an ideal risk measure in portfolio theory. *International Journal of Theoretical and Applied Finance 11*(01): 19–54 .

Sklar, M. 1959. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris* 8: 229–231 .

Song, X. and A. Taamouti. 2021. Measuring Granger causality in quantiles. *Journal of Business & Economic Statistics 39*(4): 937–952 .

Taylor, J.W. 2000. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting 19*(4): 299–311 .

Xu, Q., K. Deng, C. Jiang, F. Sun, and X. Huang. 2017. Composite quantile regression neural network with applications. *Expert Systems with Applications* 76: 129–139 .

Yeh, I.C. and W.L. Cheng. 2010. First and second order sensitivity analysis of MLP. *Neurocomputing 73*(10-12): 2225–2233 .

Zhang, Z., S. Zohren, and S. Roberts. 2020. Deep learning for portfolio optimization. *The Journal of Financial Data Science 2*(4): 8–20 .

Zheng, S. 2011. Gradient descent algorithms for quantile regression with smooth approximation. *International Journal of Machine Learning and Cybernetics 2*(3): 191–207 .

Zou, H. and M. Yuan. 2008. Composite quantile regression and the oracle model selection theory. *The Annals of Statistics 36*(3): 1108–1126 .

Zurada, J.M., A. Malinowski, and I. Cloete 1994. Sensitivity analysis for minimization of input data dimension for feedforward neural network. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, Volume 6, pp. 447–450. IEEE.