Expert Systems With Applications 233 (2023) 120934

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A novel recurrent neural network based online portfolio analysis for high frequency trading

Xinwei Cao [a], Adam Francis [b], Xujin Pu [a], Zenan Zhang [a], Vasilios Katsikis [c], Predrag Stanimirovic [d], Ivona Brajevic [e], Shuai Li [f,*]

[a] School of Business, Jiangnan University, Wuxi, China
[b] College of Engineering, Swansea University, Swansea, UK
[c] Department of Economics, Division of Mathematics and Informatics, National and Kapodistrian University of Athens, Athens, Greece
[d] Faculty of Science and Mathematics, University of Nis, Nis, Serbia
[e] Faculty of Applied Management, Economics and Finance, University Business Academy, Belgrade, Serbia
[f] Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland

ARTICLE INFO

Keywords:
Recurrent neural network
Pareto frontier
Portfolio analysis
Markowitz model
Time-varying problem

ABSTRACT

The Markowitz model, a Nobel Prize winning model for portfolio analysis, paves the theoretical foundation in finance for modern investment. However, it remains a challenging problem in the high frequency trading (HFT) era to find a more time efficient solution for portfolio analysis, especially when considering circumstances with the dynamic fluctuation of stock prices and the desire to pursue contradictory objectives for less risk but more return. In this paper, we establish a recurrent neural network model to address this challenging problem in runtime. Rigorous theoretical analysis on the convergence and the optimality of portfolio optimization are presented. Numerical experiments are conducted based on real data from Dow Jones Industrial Average (DJIA) components and the results reveal that the proposed solution is superior to DJIA index in terms of higher investment returns and lower risks.

1. Introduction

Investment in stock markets is one of the most popular means for both retail and institutional investors in our daily lives. One of the central problems is the determination of particular stocks to short/long for maximum profit. From a financial perspective, it is a game between risk and return. The greediness of humans always push for an answer to the question: how to allocate money to different stocks, i.e., the optimal portfolio of stocks, so that the risk is minimized while the return is maximized. Markowitz's pioneering work modeled stock prices as a random variable and characterized risks and returns by statistical variance and mean respectively. This formal framework provided the first practical but reasonably simple model for practitioners to make rational decisions for financial investment. Markowitz, as the inventor of this insightful model, received the 1990 Nobel prize in economics (Markowitz, 1991). This model for portfolio analysis paved the foundation for modern portfolio theory (MPT) and constitutes as a supporting pillar of modern finance (Kim & Francis, 2013; Omisore et al., 2011).

Although great success has been achieved for portfolio analysis with the birth of Markowitz model, the demand for timely decision making has significantly increased especially in recent years with the advancement of high frequency trading (HFT), which combines powerful computing servers and the fastest Internet connection to trade at extremely high speeds. This demand poses new challenges to portfolio solvers for real-time processing in the face of time-varying parameters. Neural networks, as one of the most powerful machine learning tools (Khan et al., 2020, 2022), has seen great progress in recent years for financial data analysis and signal processing (Adamu, 2019; Gao & Su, 2020; Khan et al., 2021; Troiano et al., 2018). Using computational methods, e.g., machine learning and data analytics, to empower conventional finance is becoming a trend widely adopted in leading investment companies (Allen et al., 2021; Goldstein et al., 2019; Rebentrost et al., 2018; Sharma et al., 2022; Tran et al., 2018; Tsantekidis et al., 2020).

* Corresponding author.
E-mail addresses: xwcao@jiangnan.edu.cn (X. Cao), a.g.francis@swansea.ac.uk (A. Francis), puyiwei@ustc.edu (X. Pu), zhzn@jiangnan.edu.cn (Z. Zhang), vaskatsikis@econ.uoa.gr (V. Katsikis), pecko@pmf.ni.ac.rs (P. Stanimirovic), ivona.brajevic@mef.edu.rs (I. Brajevic), shuai.li@oulu.fi (S. Li).

https://doi.org/10.1016/j.eswa.2023.120934
Received 4 August 2022; Received in revised form 30 June 2023; Accepted 30 June 2023
Available online 8 July 2023

Since the establishment of Markowitz model for portfolio analysis, it has been an active research direction to develop variant models for performance enhancement. In Davis and Norman (1990) and Zhang et al. (2012), the authors point out the limitation of original Markowitz model for not considering the transaction cost and they develop more realistic models with this factor taken into account. For some investors, one preference is to restrict the number of selected stocks in the portfolio into an understandable and interpretable amount. The authors in Chang et al. (2000) and Gao et al. (2015) model this preference as a cardinality constraint in their optimization formulation. However, the cardinality constraint is non-convex and conventional solvers usually get stuck at the local optima. To find optimal solutions, some researchers relax the cardinality to L_1 constraint, one convex approximation of cardinality, and leverage recent progress on sparsity optimization as a solution (Kremer et al., 2020). In the mean while, some researchers explore the solution using evolutionary computation methods with global optimization capability, e.g., genetic algorithm (Chen et al., 2019; Yaman & Dalkilic, 2021), particle swarm optimization (Kaucic, 2019; Raei & Alibeiki, 2010; Zhu et al., 2011), beetle antennae search (Katsikis et al., 2020; Khan et al., 2019, 2023; Medvedeva et al., 2021). However, these types of methods require great computational support and may not be suitable for time-varying scenarios. Other solutions include the use of fuzzy logics (Galankashi & Mokhatab, 2020), and neural networks (Imajo et al., 2021) but all suffer from expensive computational cost and can hardly be used in real-time high frequency trading.

As a special type of neural network, recurrent neural networks with feedback connected neurons are naturally suitable for time varying data processing. In Guo and Zhang (2012) and Zhang and Ge (2005), a recurrent neural network is presented for time-varying matrix inversion. A general class of non-constrained optimization problems, including linear equation set, can be converted into matrix inversion and therefore can be efficiently solved using this model. Its discrete-time version is explored in Li et al. (2021), Liao et al. (2016), Ma and Guo (2020), Shi and Zhang (2020) and Zhang et al. (2021). In Chen et al. (2020) and Zhang et al. (2020, 2022), researchers leverage deliberate design of activation functions for the speedup of convergence to finite time, which further enhances the real time processing capability of recurrent neural networks. Further investigations include the extension to complex-valued number dynamical signal processing (Xiao et al., 2021), noise-robust neural network derivation (Li et al., 2020; Xiao et al., 2022), model-free situations (Zhang et al., 2018), robot arm motion control (Li et al., 2018), mobile robot trajectory planning (Chen et al., 2021), multiple robot coordination (Li et al., 2017), Sylvester equation solving (Zhang et al., 2019), non-stationary quadratic programs (Qi et al., 2022) and rank-deficit problem solving (Shi et al., 2022), as comprehensively surveyed in Jin et al. (2017).

Motivated by the success of recurrent neural networks in processing time-varying problems, and the demand for online analysis of portfolios in a high frequency trading era, in this paper we present a novel recurrent neural network customized for portfolio optimization, with the consideration of transaction cost and multiple objectives. This neural network is proved to find the solution corresponding to the optimum portfolio. This method provides a superior solution to both retail and institutional investors than simply buying market index. In addition, the method can also output the Pareto efficient frontier, which is of great importance for practitioners when making financial decisions. The main contributions of this paper are summarized as follows.

(1) Conventional solutions are suitable for static problems without time-varying parameters. In the high frequency trading era, many applications involve time-dependent factors. The proposed method is suitable for solving the optimal portfolio online under time-varying parameters.
(2) Compared to evolutionary optimization based solutions, the proposed method does not require high computational power. This feature enhances the suitability of the proposed model for high frequency trading.
(3) The optimality and global convergence of the proposed model are guaranteed in theory. The model can deliver important financial indices, including the optimal portfolio, expected return, and Pareto frontier.
(4) To the best of our knowledge, this is the first recurrent neural network model developed for portfolio analysis without the potential risk of local optima in the final solution.

The rest of the paper is organized as follows. In Section 2, the portfolio analysis problem is described. In Section 3, the neural network is designed to solve the formulated problem and theoretical analysis is illustrated. Then, experimental validations based on real stock data are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Problem description

Portfolio analysis refers to the determination of the optimal distribution of investments to different stocks so that less risk and more return can be achieved. In this section, we present the formulation of portfolio analysis.

2.1. Stock return

Consider a total of n stocks named S_i for the ith one. Mathematically, the return of the ith stock at time t, denoted as $r_i(t)$, is a random variable up to fluctuation. The following statistical characteristics are defined for $r_i(t)$:

$$\begin{aligned} E(r_i(t)) &= \mu_i(t), \\ Cov(r_i(t), r_j(t)) &= \sigma_{ij}(t) \text{ with } \sigma_{ij}(t) = \sigma_{ji}(t), \end{aligned} \tag{1}$$

where $E(\cdot)$ computes the mean value of a random variable, and $Cov(\cdot,\cdot)$ for the covariance of two variables, $\mu_i(t)$ is the mean value of r_i, σ_{ij} is the covariance of r_i and r_j. $\sigma_{ij}(t)$ reveals the inter-dependency of the two stocks S_1 and S_2 in their return. The value of $\sigma_{ij}(t)$ could be greater than 0 when their returns are positively correlated, or less than 0 when negatively correlated. For example, the stock trend of Amazon and Ebay are positively correlated most of the time due to their common nature as online consumer markets, while crude oil prices and airline stocks are usually observed to be negatively correlated due to the impact of crude oil price on airline profits. In a compact form, Eq. (1) can be written as

$$E(r(t)) = \mu(t), Var(r(t)) = \Sigma(t), \tag{2}$$

with

$$\mu(t) = \begin{bmatrix} \mu_1(t) \\ \mu_2(t) \\ \vdots \\ \mu_n(t) \end{bmatrix}, \Sigma(t) = \begin{bmatrix} \sigma_{11}(t) & \sigma_{12}(t) & \cdots & \sigma_{1n}(t) \\ \sigma_{21}(t) & \sigma_{22}(t) & \cdots & \sigma_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1}(t) & \sigma_{n2}(t) & \cdots & \sigma_{nn}(t) \end{bmatrix}, \tag{3}$$

where the variance matrix $\Sigma(t) \in \mathbb{R}^{n\times n}$ is positive semi-definite.

2.2. Mean return VS. Risk

The decision variable in Markowitz's portfolio analysis is the distribution of investments. Suppose one investor holds one dollar and a wise decision is to select a proper fraction $w_i(t)$ to buy S_i for $i = 1, 2, \dots, n$ so that the overall return is maximized while the risk minimized. Two extremes are evident: it is the best decision to buy stock S_{i^*} for $i^* = \text{argmax}\{r_1(t), r_2(t), \dots, r_n(t)\}$, i.e., the most profit stock with maximum statistic mean return, if an investor wants maximize mean return without taking the potential risk into account; oppositely, a rational investor distribute the money over different stocks to reduce risks, following the age-old philosophy 'do not put all your eggs in one basket'. For one dollar overall investment with $w_i(t) \geq 0$ for stock S_i, the overall amount $\sum_1^n w_i(t) = 1$, with the matrix form as

$$w^{\mathrm{T}}(t)\mathbf{1} = 1, \tag{4}$$

where $w^{\mathrm{T}}(t) = [w_1(t), w_2(t), \ldots, w_n(t)]$ is the weight vector and is also the decision variable of portfolio analysis, and $\mathbf{1} \in \mathbb{R}^n$ is a n dimensional vector with all entry 1. With this portfolio of stocks, the return $p(t)$ is therefore computed as

$$p(t) = w^{\mathrm{T}}(t)r(t). \tag{5}$$

Apparently, the portfolio return $p(t)$ is also a random variable. We can compute its mean and variance respectively as below:

$$\begin{aligned} E(p(t)) &= w^{\mathrm{T}}(t)E(r(t)) = w^{\mathrm{T}}(t)\mu(t), \\ Var(p(t)) &= Var(w^{\mathrm{T}}(t)r(t)) = w^{\mathrm{T}}(t)\Sigma(t)w(t). \end{aligned} \tag{6}$$

2.2.1. Return maximization

Quantitatively, the maximization of the mean return, without considering potential risks, results in the following formulation of an optimization problem

$$\begin{aligned} &\max_{w(t)} w^{\mathrm{T}}(t)\mu(t) \\ &\quad s.t.\, w^{\mathrm{T}}(t)\mathbf{1} = 1, \\ &\qquad 0 \le w_i(t) \le 1, \forall i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

The closed-form optimal solution can be readily solved as

$$w(t) = e_{i^*}(t) \text{ for } i^* = \operatorname{argmax}\{r_1(t), r_2(t), \ldots, r_n(t)\}, \tag{8}$$

where $e_{i^*}(t) \in \mathbb{R}^n$ is a n dimensional vector with all entries being 0 except the i^*th entry being 1. Solution (8) represents the investment of all money in the most profitable stock S_{i^*} with the highest mean return $r_{i^*}(t)$.

2.2.2. Risk minimization

In contrast, the minimization of risks, characterized by the variance of the portfolio $p(t)$, can be described by the following optimization

$$\begin{aligned} &\min_{w(t)} w^{\mathrm{T}}(t)\Sigma(t)w(t) \\ &\quad s.t.\, w^{\mathrm{T}}(t)\mathbf{1} = 1, \\ &\qquad 0 \le w_i(t) \le 1, \forall i = 1, 2, \ldots, n. \end{aligned} \tag{9}$$

Problem (9) can also be solved in closed form by noticing the eigenvalue inequality $w^{\mathrm{T}}(t)\Sigma(t)w(t) \ge \lambda_1(t)w^{\mathrm{T}}(t)w(t)$ for any $w(t) \neq 0$, where $\lambda_1(t)$ is the smallest eigenvalue of $\Sigma(t)$. For $w^{\mathrm{T}}(t)\Sigma(t)w(t) \ge \lambda_1(t)w^{\mathrm{T}}(t)w(t)$, equality holds when $w(t)$ equals $x_1(t)$, where $x_1(t)$ is the eigenvector of $\Sigma(t)$ corresponding to its smallest eigenvalue $\lambda_1(t)$. Accordingly, the solution of (9) can be written as

$$w(t) = \frac{x_1(t)}{\|x_1(t)\|}, \; x_1(t) \neq 0 \tag{10}$$

where $x_1(t) \in \mathbb{R}^n$ is the corresponding eigenvector of $\Sigma(t)$ to its smallest eigenvalue $\lambda_1(t)$. Note that $x_1(t)$ in the denominator is for normalization so that entries of $w(t)$ sum to one. Solution (10) implies that the risk-minimum portfolio can be reached with the distribution of money over all stocks according to the eigenvector of the return variance matrix $\Sigma(t)$.

2.3. Transaction cost and multi-objective optimization

2.3.1. Transaction cost

In practice, the transaction of stocks introduces additional expenses. Suppose the transaction cost rate is $c(t) \in \mathbb{R}^n$ with $c_i(t)$ being the *ith* element. For the investment of amount $w_i(t)$ to stock S_i, the individual transaction cost is $w_i(t)c_i(t)$ and the total transaction cost of all stocks in the portfolio is $\sum_{i=1}^n w_i(t)c_i(t) = w^{\mathrm{T}}(t)c(t)$, resulting in the replacement of Eq. (4) by the following

$$w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1. \tag{11}$$

2.3.2. Multi-objective optimization

Emotional investors desire the minimum risk as well as the maximum mean return, which leads to the following formulation of the problem

$$\begin{aligned} &\min_{w(t)} w^{\mathrm{T}}(t)\Sigma(t)w(t), \\ &\max_{w(t)} w^{\mathrm{T}}(t)\mu(t), \\ &\quad s.t.\, w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1, \\ &\qquad 0 \le w_i(t) \le 1, \forall i = 1, 2, \ldots, n. \end{aligned} \tag{12}$$

Actually, the first and the second objectives in (12) cannot be achieved simultaneously by a single decision in most practical situations, implying the non-feasibility of the demand by emotional investors. Intuitively, the portfolio $w(t)$ that achieves the first objective in (12), i.e., the one with minimum risk, usually is not identical to the portfolio that maximizes the mean return, which is the goal of the second objective in (12).

2.4. Problem reformulation

We present two reformulations for solving problem (12).

2.4.1. Risk minimization with desired return

Rather than pursuing the maximization of the mean return, a reasonable goal is to set the desired mean return of portfolio $p(t)$ as a proper value $\mu_d(t)$ and minimize the risk under this constraint, leading to the following constrained optimization problem:

$$\begin{aligned} &\min_{w(t)} w^{\mathrm{T}}(t)\Sigma(t)w(t), \\ &\quad s.t.\, w^{\mathrm{T}}(t)\mu(t) = \mu_d(t), \\ &\qquad w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1, \\ &\qquad 0 \le w_i(t) \le 1, \forall i = 1, 2, \ldots, n. \end{aligned} \tag{13}$$

Note that it returns exactly the same solution if we replace the first equation constraint $w^{\mathrm{T}}(t)\mu(t) = \mu_d(t)$ in (13) by $w^{\mathrm{T}}(t)\mu(t) \ge \mu_d(t)$ as the minimization of the objective function $w^{\mathrm{T}}(t)\Sigma(t)w(t)$ drives $w(t)$ to reduce in values as much as possible until the equality holds.

2.4.2. Optimization in Pareto sense

Pareto optima is defined as an alterative to address the multi-objective dilemma. It aims to find such decisions where the change of it can never simultaneously improve/keep all the objectives. Usually, Pareto optimum returns a set of solutions. Markowitz adopted the optimality in Pareto Sense to find the efficient frontier, i.e., the Pareto front. According to Pareto optimization theory (Miettinen, 2012), the set of solutions in Pareto sense can be obtained by solving the optimization below

$$\begin{aligned} &\min_{w(t)} w^{\mathrm{T}}(t)\Sigma(t)w(t) - k(t)w^{\mathrm{T}}(t)\mu(t), \\ &\quad s.t.\, w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1, \\ &\qquad 0 \le w_i(t) \le 1, \forall i = 1, 2, \ldots, n, \end{aligned} \tag{14}$$

with $k(t) \ge 0$ scanning across all feasible values.

Optimization (14) can be used to find the optimal portfolio set in Pareto sense, i.e., the Markowitz efficient frontier as named in fiance, by enumerating possible values of $k(t)$. Besides, there also exists some underlying equivalence between (14) and (13). For a given value of $k(t)$ in (14), it is always possible to increase $k(t)$ until $w^{\mathrm{T}}(t)\mu(t) = \mu_d(t)$ if $w^{\mathrm{T}}(t)\mu(t) \le \mu_d(t)$ for a given $\mu_d(t)$, and vice versa. This implies that (13) can be equivalently solved with (14) by choosing a proper value of $k(t)$ for a given value of $\mu_d(t)$. Without losing generality, we therefore focus on the solution of the general optimization form (14) in this paper.

3. Recurrent neural networks for real time portfolio analysis

In this section, we propose a recurrent neural network based approach to solve the time-varying portfolio optimization (14) and then provide theoretical analysis for the result.

3.1. Derivation of the recurrent neural network

As (14) includes constraints in the form of both equalities and inequalities, we first construct the following Lagrangian function to effectively deal with the constraints

$$L(w(t),\lambda(t)) = w^{\mathrm{T}}(t)\Sigma(t)w(t) - k(t)w^{\mathrm{T}}(t)\mu(t) + \lambda(t)(w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) - 1), \tag{15}$$

where $\lambda(t) \in \mathbb{R}$ is a Lagrangian multiplier associated with the equation $w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1$ constraint. Define a convex set

$$\Omega = \{w(t) \in \mathbb{R}^n, w_i(t) \le 1, \forall i = 1, 2, \dots, n\} \tag{16}$$

to capture the inequality constraint. According to optimization theory (Ruszczynski, 2006), the following min–max optimization reaches the same solution to (14)

$$\min_{w(t)}\max_{\lambda(t)} L(w(t),\lambda(t)), \quad s.t.\ w \in \Omega. \tag{17}$$

In light of the Karush–Kuhn–Tucker (KKT) condition (Ruszczynski, 2006), the solution of (17) can be written as

$$0 \in \frac{\partial L(w(t),\lambda(t))}{\partial w(t)} + N_\Omega(w(t)),\ 0 = \frac{\partial L(w(t),\lambda(t))}{\partial \lambda(t)} \tag{18}$$

where $N_\Omega(w(t))$ is the normal cone of set Ω at $w(t)$. The normal cone expressed condition (18) can be equivalent converted to set projection descriptions as below

$$P_\Omega\left(w(t) - \frac{\partial L(w(t),\lambda(t))}{\partial w(t)}\right) = w(t), \quad \frac{\partial L(w(t),\lambda(t))}{\partial \lambda(t)} = 0, \tag{19}$$

where $P_\Omega(\cdot)$ is the projection operator to set Ω. Expression (19) is a set of nonlinear equations and still cannot be solved effectively. We therefore construct the following recurrent neural network, whose equilibrium point is identical to the solution of (19), for real time solution

$$\epsilon\dot{w}(t) = -w(t) + P_\Omega\left(w(t) - \frac{\partial L(w(t),\lambda(t))}{\partial w(t)}\right), \quad \epsilon\dot{\lambda}(t) = \frac{\partial L(w(t),\lambda(t))}{\partial \lambda(t)}, \tag{20}$$

with $\epsilon > 0$ being a scaling factor. The substitution of Expression (15) into Eq. (20) results in

$$\begin{aligned}\epsilon\dot{w}(t) = &-w(t) + P_\Omega(w(t) - 2\Sigma(t)w(t) + k(t)\mu(t) \\ &- \lambda\mathbf{1} - \lambda c(t)), \\ \epsilon\dot{\lambda}(t) = &\, w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) - 1,\end{aligned} \tag{21}$$

which yields the recurrent neural network to facilitate online solution of portfolio analysis. We have the following remark for the derived recurrent neural network (21).

Remark 1. The recurrent neural network (21) is designed for the solution of (14). The projection term $P_\Omega(\cdot)$ in (20) enforces the inequality constraint (the 3rd line) in (21), the $\lambda(t)$ dynamics enforces the equality constraint (the 2rd line) and the $w(t)$ dynamics enforces the goal to minimize the objective function (the 1st line). The above three factors work in coordination to enforce the solution of (21) in a recurrent way through the dynamical evolution of $w(t)$ and $\lambda(t)$.

3.2. Theoretical analysis

In this part, we provide two main theoretical results of the designed recurrent neural network (21) to pave the foundation for the effectiveness of using (21) for the solution of (14). The derived neural network is generally an ordinary differential equation. Our first result is on its equilibrium point and the second result is on its global convergence.

3.2.1. Equilibrium point optimality

About the equilibrium point, we have the following theorem.

Theorem 1. *The equilibrium point $(w(t), \lambda(t))$ of recurrent neural network (21) is identical to the optimal solution of (14).*

Proof. The equilibrium point of (21) is obtained as $0 = -w(t) + P_\Omega(w(t) - 2\Sigma(t)w(t) + k(t)\mu(t) - \lambda\mathbf{1} - \lambda c(t))$ and $0 = w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) - 1$. This can be further written as $0 = -w(t) + P_\Omega(w(t) - \partial L(w(t),\lambda(t))/\partial w(t))$ and $0 = \partial L(w(t),\lambda(t))/\partial\lambda(t)$ with the definition of $L(w(t),\lambda(t)) = w^{\mathrm{T}}(t)\Sigma(t)w(t) - k(t)w^{\mathrm{T}}(t)\mu(t) + \lambda(w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) - 1)$. Note that the above defined function $L(w(t),\lambda(t))$ is convex relative to $w(t)$ owing to the positive semi-definite property of the variance matrix $\Sigma(t)$, and $L(w(t),\lambda(t))$ is linear relative to $\lambda(t)$. Therefore, the KKT condition implies that the equilibrium point is the only solution of $\min_{w(t)\in\Omega}\max_{\lambda(t)} L(w(t),\lambda(t))$. As $L(w(t),\lambda(t))$ can also be recognized as a Lagrangian function with $w(t)$ being the decision variable, $\lambda(t)$ being the multiplier, and $w^{\mathrm{T}}(t)\Sigma(t)$ $w(t) - k(t)w^{\mathrm{T}}(t)\mu(t)$ being the objective to be minimized. Accordingly, we obtain the following equivalent optimization problem

$$\min_{w(t)\in\Omega} w^{\mathrm{T}}(t)\Sigma(t)w(t) - k(t)w^{\mathrm{T}}(t)\mu(t), \quad s.t.\ w^{\mathrm{T}}(t)\mathbf{1} + w^{\mathrm{T}}(t)c(t) = 1. \tag{22}$$

This is identical to optimization (14) by recalling the definition of set Ω. Note that all the above procedures are conducted under equivalent conversion. Therefore, the conclusion is both sufficient and necessary. This concludes the proof. □

3.2.2. Global convergence

We have so far proved that the equilibrium point of the recurrent neural network is identical to the optimal solution. We can readily declare that the designed neural model is able to solve the formulated optimization problem if the neural dynamics converges globally to its equilibrium point whatever the initial state is started with.

About the global convergence, we have the theorem below.

Theorem 2. *Recurrent neural network (21) converges globally to the optimal solution of problem (14).*

Proof. As we have proved that the equilibrium point of recurrent neural network (21) is identical to the optimal solution of problem (14). We thus only need to prove the global convergence of (21) to its equilibrium point.

Define an extended variable $x^{\mathrm{T}}(t) = [w^{\mathrm{T}}(t), \lambda(t)]$ and an extended set $\bar{\Omega} = \{x^{\mathrm{T}}(t) = [w^{\mathrm{T}}(t), \lambda(t)], w(t) \in \Omega, \lambda(t) \in \mathbb{R}\}$ and the neural dynamics (21) can be expressed in a compact form as

$$\epsilon\dot{x}(t) = -x(t) + P_{\bar{\Omega}}(x(t) - F(x(t))) \tag{23}$$

where

$$F(x(t)) = J(t)x(t) - b(t) \tag{24}$$

with matrix $J(t) \in \mathbb{R}^{(n+1)\times(n+1)}$ and vector $b(t) \in \mathbb{R}^{n+1}$ defined as

$$J(t) = \begin{bmatrix} 2\Sigma(t) & \mathbf{1} + c(t) \\ -\mathbf{1}^{\mathrm{T}} - c^{\mathrm{T}}(t) & 0 \end{bmatrix},\ b(t) = \begin{bmatrix} k(t)\mu(t) \\ -1 \end{bmatrix} \tag{25}$$

with the following property for $J(t)$

$$J(t) + J^{\mathrm{T}}(t) = \begin{bmatrix} 4\Sigma(t) & 0 \\ 0 & 0 \end{bmatrix} \succcurlyeq 0, \tag{26}$$

where '$\succcurlyeq$' means the left side is a positive semi-definite matrix. Eq. (26) holds by noticing the positive semi-definiteness of the variance matrix. Therefore,

$$\frac{\partial F(x(t))}{\partial x(t)} + \frac{\partial^{\mathrm{T}} F(x(t))}{\partial x(t)} = J(t) + J^{\mathrm{T}}(t) \succcurlyeq 0. \tag{27}$$

System (23) is in the canonical form of projected dynamical systems and it is globally convergent to its equilibrium point provided (27)

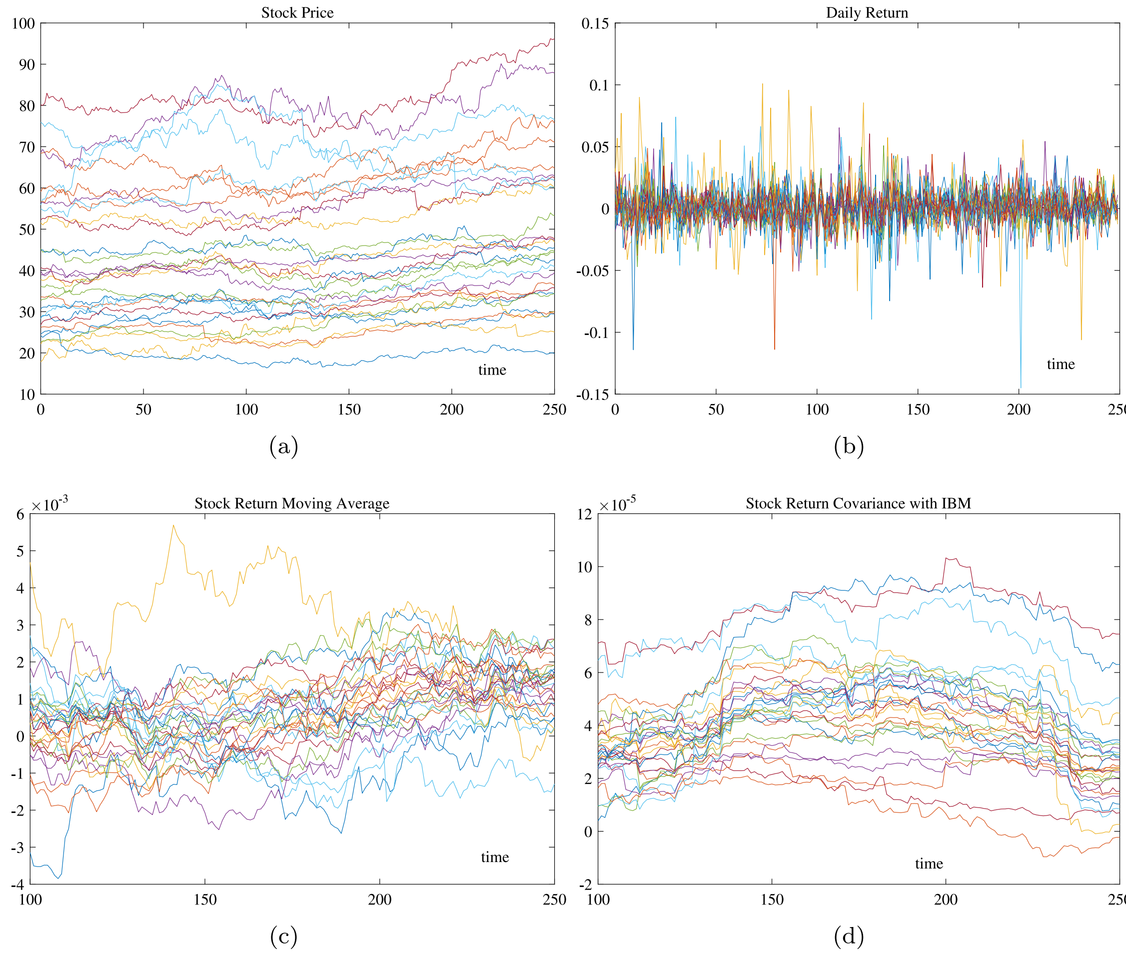

Fig. 1. Data used in the experimental validation. (a) Stock price of DJI components for 250 effective days from Jan. 3rd 2006 to Dec. 29th 2006. (b) Daily return calculated from the stock price. (c) Mean return calculated from the moving average based on data from 100 latest days. (d) Covariance of DJI companies with IBM based on data from 100 latest days.

holds, as proved in Xia (2004) and discussed in Mohammed and Li (2015). We therefore conclude the global convergence of (21) to its equilibrium point and thus complete the proof after combining with the fact that the equilibrium point of (23) is identical to the optimal solution of (14) as obtained in Theorem 1. □

4. Numerical validation on DJI components

Dow Jones Industrial Average (DJIA) is one of the most widely adopted index reflecting the overall trend of stock markets in USA. It is obtained by price-averaging the stock of 30 prominent companies listed on New York Stock Exchanges. It is usually a good choice to invest on DJIA compared to the investment to an individual stock. In this section, we apply the developed method to the portfolio of the above 30 stocks and show the superiority of the resulting solution using our method to investment on DJIA.

4.1. Experimental setup

4.1.1. Data set

In our experiment, we consider the 30 DJI component stocks, including Alcoa Corporation (AA), American International Group, Inc. (AIG), American Express Company (AXP), The Boeing Company (BA), Citigroup Inc. (C), Caterpillar Inc. (CAT), DuPont de Nemours, Inc. (DD), The Walt Disney Company (DIS), General Electric Company (GE), General Motors Company (GM), The Home Depot, Inc. (HD), Honeywell International Inc. (HON), HP Inc. (HPQ), International Business Machines Corporation (IBM), Intel Corporation (INTC), Johnson & Johnson (JNJ), JPMorgan Chase & Co. (JPM), The Coca-Cola Company (KO), McDonald's Corporation (MCD), 3M Company (MMM), Altria Group, Inc. (MO), Merck & Co., Inc. (MRK), Microsoft Corporation (MSFT), Pfizer Inc. (PFE), The Procter & Gamble Company (PG), AT&T Inc. (T), Raytheon Technologies Corporation (UTX), Verizon Communications Inc. (VZ), Walmart Inc. (WMT) and Exxon Mobil Corporation (XOM), from Jan. 3rd 2006 to Dec. 29th 2006 published online for a total of 250 days with data available.

4.1.2. Data pre-processing

Fig. 1(a) shows the stock price $v(t)$ of the 30 stocks over the 250 days. The stock price is converted to daily return $r(t)$ defined as $r(t) = (v(t) - v(t-1))/v(t)$, as shown in Fig. 1(b). To accommodate portfolio calculation, the statistical characteristics for portfolio need to be constructed. We use the numerical mean $\hat{\mu}(t) = \sum_{\tau=t-m+1}^{t} r(\tau)/m$ and the numerical variance $\hat{\Sigma}(t) = \sum_{\tau=t-m+1}^{t} (r(\tau) - \hat{\mu}(t))(r(\tau) - \hat{\mu}(t))^{\mathrm{T}}/(m-1)$ based on daily return in the past m days as the estimates of mean return $\mu(t)$ and return variance $\Sigma(t)$ respectively. Linear interpolation is employed to convert $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ from discrete time in daily basis to continuous time. See Fig. 1(c) for the corresponding $\hat{\mu}(t)$ with m chosen as $m = 100$ for the considered data. As $\hat{\Sigma}(t)$ is a 30×30 matrix in this case which includes a total of 900 values for every time instance, for the

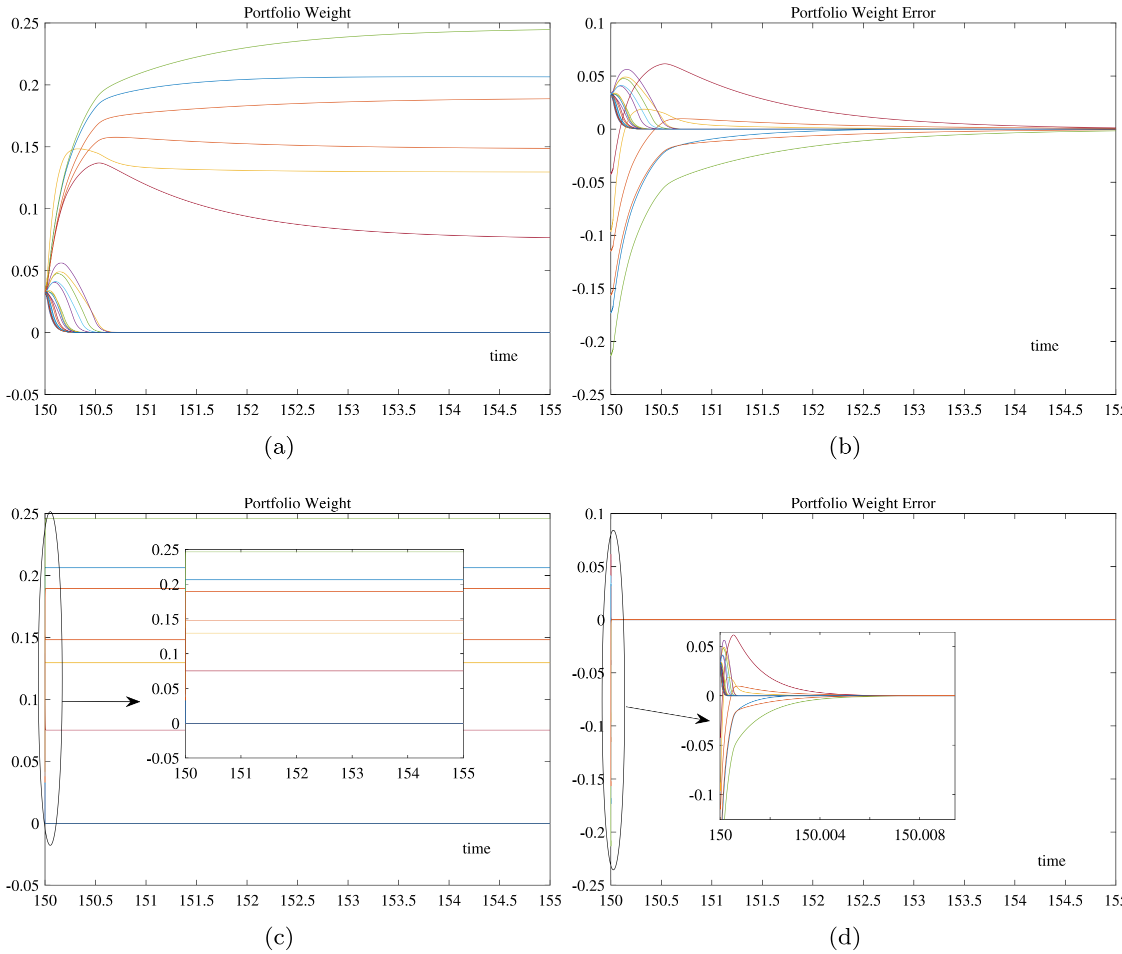

Fig. 2. Recurrent neural network based portfolio analysis for problems with fixed parameters. (a) The time profile of the calculated optimal portfolio weight using the proposed recurrent neural network with $\epsilon = 5.0 \times 10^{-2}$. (b) The time profile of the portfolio weight error using the proposed recurrent neural network with $\epsilon = 5.0 \times 10^{-2}$. (c) The time profile of the calculated optimal portfolio weight using the proposed recurrent neural network with $\epsilon = 5.0 \times 10^{-5}$. (d) The time profile of the portfolio weight error using the proposed recurrent neural network with $\epsilon = 5.0 \times 10^{-5}$.

convenience of data visualization, we instead plot in Fig. 1(d) the time history of the row in $\hat{\Sigma}(t)$ corresponding to the covariance of IBM with all 30 companies, rather than showing the 900 curves. The transaction cost rate $c(t)$ in (14) is defined as a vector in our formulation. In practice, stocks mostly have identical transaction cost rate with fixed value, i.e., $c(t) = c_0\mathbf{1}$ with c_0 being a constant set at $c_0 = 0.5\%$ for the numerical validation.

4.2. Portfolio with fixed parameters

To verify the correctness of the presented recurrent neural network, we first consider the typical time-invariant situation which is solvable by conventional methods. We consider the solution of (14) at time $t = 150$ based on statistics, i.e., the estimated mean return $\hat{\mu}(t) = \hat{\mu}$ and the estimated return variance $\hat{\Sigma}(t) = \hat{\Sigma}$, from the past $m = 100$ days. As this is a static linearly constrained quadratic problem and its solution can be solved using Matlab routine $quadprog()$. We choose $k(t) = 0.1$ and $c(t) = c_0\mathbf{1}$ with $c_0 = 0.5\%$ in (14). With the initial value set as $1/n$ for each dimension of $x(t)$ and 0 for $\lambda(t)$, the time evolution of the computed optimal portfolio $w(t)$, as the output of the proposed recurrent neural network with the selection of the scaling factor $\epsilon = 5.0\times10^{-2}$, is shown in Fig. 2(a). As observed, $w(t)$ varies with time and gradually converges. Its error in comparison with the solution computed using function $quadprog()$ in Matlab is shown in Fig. 2(b). Note that parameter ϵ scales the time horizon and speed-up of the convergence can be achieved by choosing a smaller value of ϵ. With the same setup, a much faster convergence can be observed in Fig. 2(c) and (d) when choosing $\epsilon = 5.0 \times 10^{-5}$. It is clear that in both cases, the portfolio calculated by the proposed neural network converges to the ideal value with time, which verifies the effectiveness of the proposed method.

4.3. Portfolio with time-varying parameters

Conventional numerical methods are suitable for static portfolio optimization problems. When applied to situation with time-varying parameters, conventional solutions still treat the problem as static and solve for all time instances, making conventional methods less efficient when parameters vary with time continuously. In contrast, the recurrent neural network developed in this paper is essentially suitable for dynamic problems. In this experiment, rather than using fixed values for $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$, we consider dynamic ones with averaged running data in the past $m = 100$ days to make runtime decision with time. As before, the transaction cost rate $c(t) = c_0\mathbf{1}$ with c_0 set at $c_0 = 0.1\%$. As shown in Section 4.2, a smaller value of ϵ can reduce the time scale of the system and speed up the convergence, we therefore choose $\epsilon = 5.0 \times 10^{-5}$. Figs. 3 and 4 show the experimental result for a time-varying parameter situation using the proposed recurrent neural network based online portfolio analysis. Note that the time-varying nature of the parameters results in the consequence that the optimal portfolio also varies with

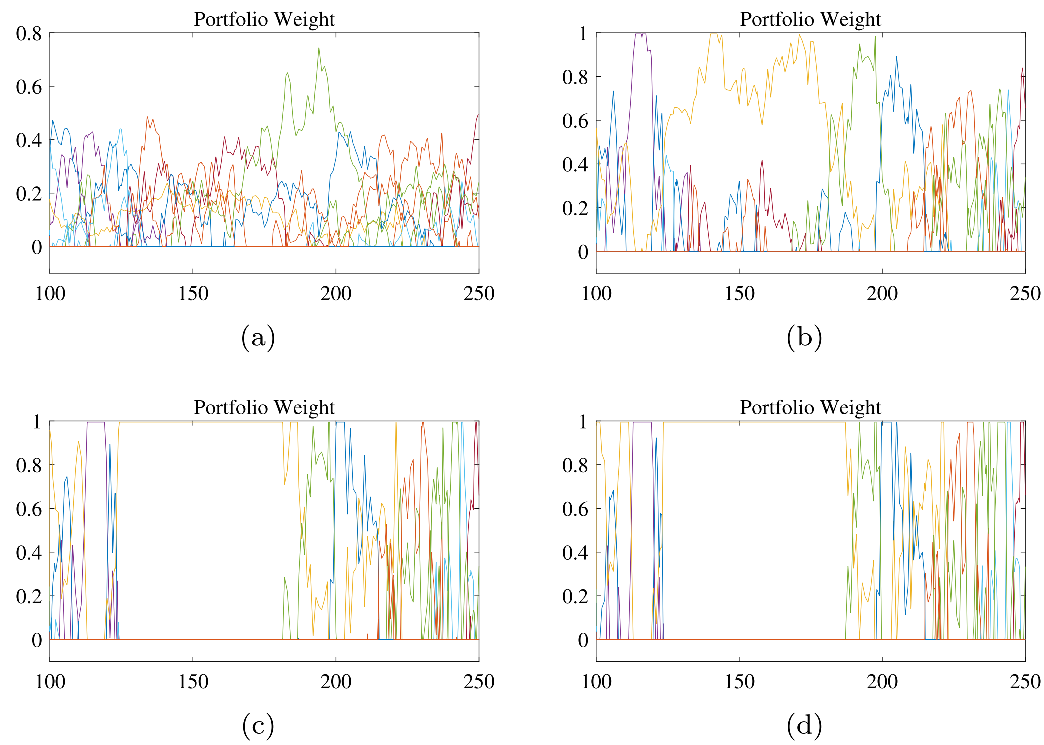

Fig. 3. The time profile of portfolio weights obtained using recurrent neural network based online portfolio analysis with different values of $k(t)$, namely $k(t) = 0.1$ for (a), $k(t) = 0.5$ for (b), $k(t) = 1.0$ for (c), and $k(t) = 1.5$ for (d).

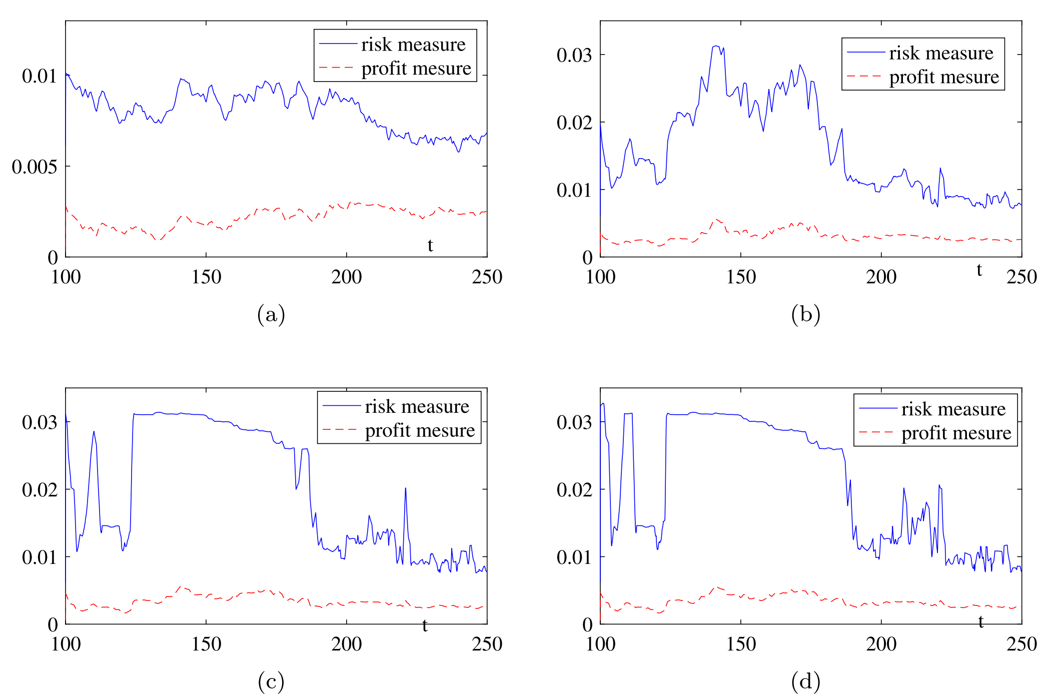

Fig. 4. The time profile of risks in terms of $(w^{\mathrm{T}}(t)\hat{\Sigma}(t)w(t))^{1/2}$ and the corresponding profit $w^{\mathrm{T}}(t)\hat{\mu}(t)$ obtained using recurrent neural network based online portfolio analysis with different values of $k(t)$, namely $k(t) = 0.1$ for (a), $k(t) = 0.5$ for (b), $k(t) = 1.0$ for (c), and $k(t) = 1.5$ for (d).

time. As shown in Fig. 3(a), with the weighting coefficient $k(t)$ set at $k(t) = 0.1$, the calculated optimal weight changes dynamically and the presented neural network always keep track of the optimal portfolio with time, allowing practitioners to make run-time decision which is especially important in high frequency transactions. The corresponding risk measure evaluated as $(w^{\mathrm{T}}(t)\hat{\Sigma}(t)w(t))^{1/2}$ and the profit measure evaluated as $w^{\mathrm{T}}(t)\hat{\mu}(t)$ are also shown in Fig. 4(a), which provides guidance to investors on the risk and the potential return at the present moment and the trend of them, implying possible opportunities for making financial decisions. It is noteworthy that the parameter $k(t)$ in model (14) serves as a tradeoff coefficient and it increases the expected profit but also increases the risk if we choose a larger $k(t)$. By increasing the value of $k(t)$ from $k(t) = 0.1$ to $k(t) = 0.5$, $k(t) = 1.0$ and further to $k(t) = 1.5$, we can evidently observe the increase of

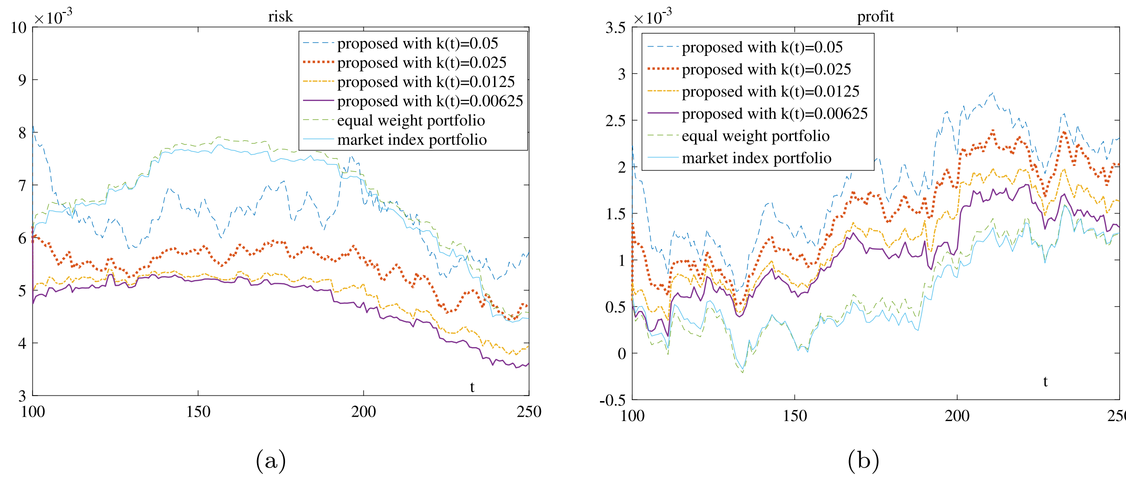

Fig. 5. Comparison of the profit and the risk of investment using different methods, including portfolio obtained using the proposed recurrent neural network (with different $k(t)$, i.e., $k(t) = 0.05$, $k(t) = 0.025$, $k(t) = 0.0125$, $k(t) = 0.00625$), the equal weight portfolio and the market index portfolio. (a) The risk profile. (b) The profit profile.

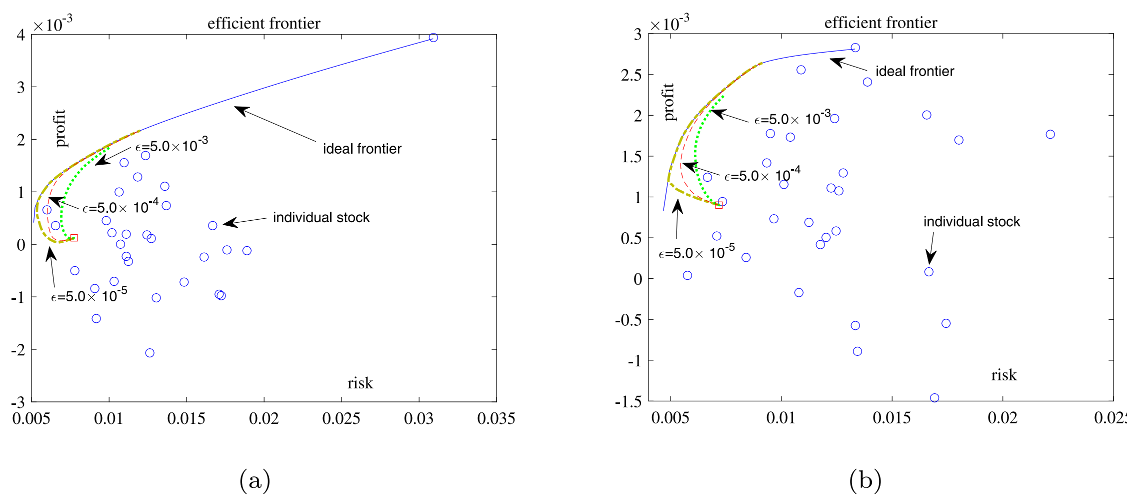

Fig. 6. Efficient frontier generated using the proposed recurrent neural network based on data at (a) $t = 150$ and (b) $t = 200$.

profits but also the increase of risks. Remarkably, as seen from Fig. 3(b) and (c) between $t = 125$ and $t = 175$, it leads to the trend to select an individual stock rather than a portfolio of them for the optimal solution with increase of $k(t)$ from 0.5 to 1.0. This aggressive behavior is due to the over-emphasize of the profit when individual stock profits are limited. Although with the same $k(t)$, for other time region than the time between $t = 125$ and $t = 175$, the change of $k(t)$ does not result in much change of the portfolio weights. This happens when the emphasize of profit can be satisfied by many individual stocks.

The advantage of the presented recurrent neural network solution can be demonstrated by two widely adopted investment strategies, namely equal weight portfolio and market index portfolio, as shown in Fig. 5. The equal weight portfolio refers to the strategy of equally distributing funds for stocks. The market index portfolio, also adopted by DJIA, is a price averaged investment scheme, which distributes funds to stocks with the price of each stocks as the weight. By properly choosing the risk and profit tradeoff coefficient $k(t)$, we can achieve higher profit but simultaneously with less risk (as evident in Fig. 5(a) for less risk with the choice of $k(t) = 0.05$, $k(t) = 0.025$, $k(t) = 0.0125$ $k(t) = 0.00625$ and higher profit as seen from Fig. 5(b)). This demonstrates the advantage of optimal portfolio in investment and the effectiveness of the proposed solution.

4.4. Dynamic tracking of efficient frontier

For two portfolios with the same profit but different risks, we can easily define the one with the lower risk as more optimal. While for two portfolios with one receiving higher risk and higher profit, and the other one receiving less risk and less profit, it is difficult to clearly define which one is superior. Pareto optimality is defined to deal with this situation. For portfolio analysis in particular, the so called Pareto efficient frontier is the optimum solution set defined in Pareto sense with both risk and profit as objectives. For each point on the frontier, we can declare that there is no other portfolio solution with the same risk but with more profit, and there is no other with the same profit but with less risk. This frontier is very useful for investors to make holistic decisions. We can easily recover the efficient frontier by setting the risk and profit tradeoff coefficient $k(t)$ as time varying in (14).

4.4.1. Efficient frontier for static investment

In this scheme, we consider the experiment with fixed mean and variance of stock returns. We consider the portfolio problem at time $t = 150$ and $t = 200$ in particular based on $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ obtained in Section 4.1 with data in the past 100 days. By running the neural network for a duration of 10 with the coefficient $k(t)$ setting as linearly

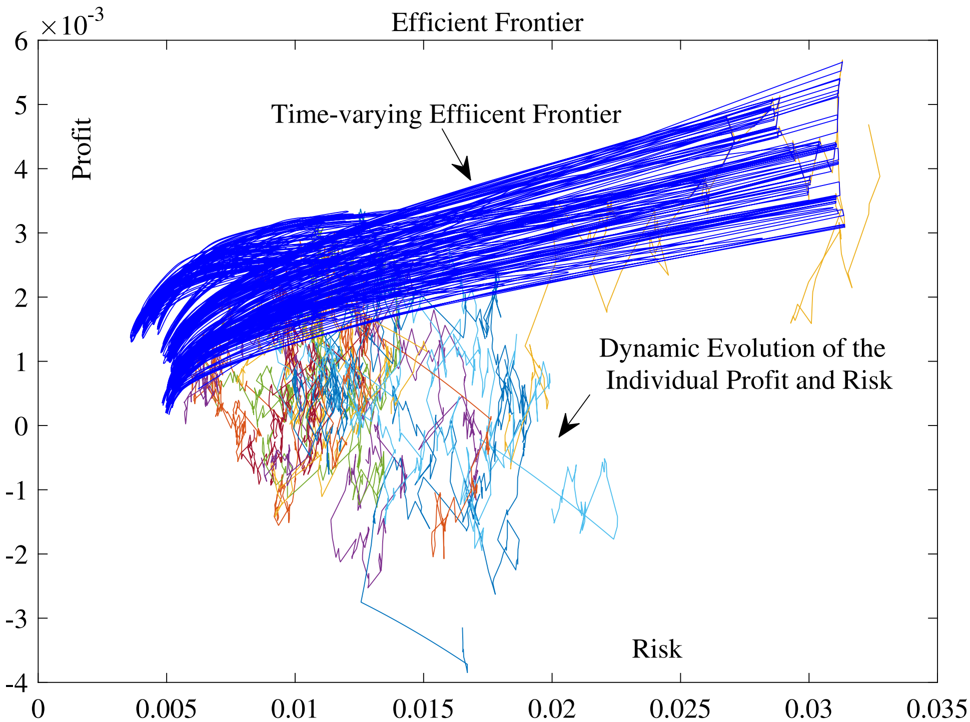

Fig. 7. Real-time efficient frontier generated using the proposed recurrent neural network for time-varying stock data.

time-varying from 0 to 2.0, we can readily obtain the Pareto efficient frontier from the portfolio output by the recurrent neural network. Fig. 6(a) shows the obtained frontier for the data at time $t = 150$. The neural network is initially set with equal weight portfolios and it evolves to track the ideal efficient frontier as observed in the figure. By choosing a larger value of ϵ, as shown in the figure, the transient can be reduced and the neural network can track the frontier faster. Similar performance can be observed for $t = 200$ in Fig. 6(b).

4.4.2. Efficient frontier for dynamic investment

In this scheme, we consider the experiment with time-varying mean and variance of stock returns. We consider the whole time range from $t = 100$ to $t = 250$ based on the estimated mean return $\hat{\mu}(t)$ and the variance of the return $\hat{\Sigma}(t)$ using data from the past 100 days. Note that both $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ vary with time in this situation and therefore, the values of $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ construct an individual efficient frontier at every moment and the frontier itself is also time-varying. While, this frontier is important to the run-time decision making in identifying trading opportunities. To capture the dynamics of this efficient frontier, we set the tradeoff weight $k(t)$ as $k(t) = (sin(2\pi t) + 1)/2$ which ranges from 0 to 1 with period 1, to capture chances in the time scale higher than 1. The output of the proposed neural network constitutes the efficient frontier as shown in Fig. 7. Note that due to the dynamic evolution of the mean return and the variance matrix, the profit and risk for each individual stocks also keep changing. In this highly dynamic situation, the proposed method can adaptively find the optimal portfolio and construct the efficient frontier, demonstrating the advantage of this method in tracking dynamic frontiers and its great potential for high frequency transactions.

5. Conclusions

To deal with time-varying data in portfolio analysis, a novel recurrent neural network is presented in this paper. The proposed method applies to two types of schemes, namely risk minimization with desired return and portfolio optimization in Pareto sense, in a unified framework. The presented method is theoretically proved for its optimality and convergence to the ideal solution of portfolio problem and extends the solution to time-varying sceneries with dynamic parameters. This method is able to find the efficient frontier of portfolios in real time. Experiments based on real stock data verify the effectiveness of the proposed method. To our best knowledge, this is the first work using dynamic neural networks for the analysis of portfolios with time-varying parameters with provable optimality.

CRediT authorship contribution statement

Xinwei Cao: Problem formulation, Paper writing. **Adam Francis:** Proof-reading, Grammar checking. **Xujin Pu:** Funding support, Methodology design. **Zenan Zhang:** Funding support, Numerical verification. **Vasilios Katsikis:** Discussion, Methodology design. **Predrag Stanimirovic:** Mathematical proof, Verification. **Ivona Brajevic:** Proof of concept, Experiment design. **Shuai Li:** Funding support, Methodology design.

Declaration of competing interest

The authors declare no conflicts of interests.

Data availability

No data was used for the research described in the article

Acknowledgments

We are grateful to the comments provided by anonymous reviewers, which helped improve the quality of this work and we are also thankful to the support by National Natural Science Foundation of China [Grant Number: 72271109] and The Ministry of Education of Humanities and Social Science Project of China [Grant Number: 22YJA630116] and the support by Jiangnan University.

References

Adamu, J. A. (2019). Advanced stochastic optimization algorithm for deep learning artificial neural networks in banking and finance industries. *Risk and Financial Management*, *1*(1), 8–28.

Allen, F., Gu, X., & Jagtiani, J. (2021). A survey of fintech research and policy discussion. *Review of Corporate Finance*, *1*, 259–339.

Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, *27*(13), 1271–1302.

Chen, C.-H., Chen, Y.-H., Lin, J. C.-W., & Wu, M.-E. (2019). An effective approach for obtaining a group trading strategy portfolio using grouping genetic algorithm. *IEEE Access*, *7*, 7313–7325.

Chen, D., Li, S., & Liao, L. (2021). A recurrent neural network applied to optimal motion control of mobile robots with physical constraints. *Applied Soft Computing*, *85*, 105–120.

Chen, D., Li, S., Lin, F.-J., & Wu, Q. (2020). New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: A finite-time and robust solution. *IEEE Transactions on Cybernetics*, *50*(6), 2651–2660.

Davis, M. H., & Norman, A. R. (1990). Portfolio selection with transaction costs. *Mathematics of Operations Research*, *15*(4), 676–713.

Galankashi, R., & Mokhatab, R. (2020). Portfolio selection: a fuzzy-anp approach. *Financial Innovation*, *6*(1), 1–34.

Gao, J., Li, D., Cui, X., & Wang, S. (2015). Time cardinality constrained mean–variance dynamic portfolio selection and market timing: A stochastic control approach. *Automatica*, *54*, 91–99.

Gao, W., & Su, C. (2020). Analysis on block chain financial transaction under artificial neural network of deep learning. *Journal of Computational and Applied Mathematics*, *380*, 112–131.

Goldstein, I., Jiang, W., & Karolyi, G. A. (2019). To fintech and beyond. *The Review of Financial Studies*, *32*(5), 1647–1661.

Guo, D., & Zhang, Y. (2012). Novel recurrent neural network for time-varying problems solving. *IEEE Computational Intelligence Magazine*, *7*(4), 61–65.

Imajo, K., Minami, K., Ito, K., & Nakagawa, K. (2021). Deep portfolio optimization via distributional prediction of residual factors. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 213–222).

Jin, L., Li, S., Liao, B., & Zhang, Z. (2017). Zeroing neural networks: A survey. *Neurocomputing*, *267*, 597–604.

Katsikis, V. N., Mourtas, S. D., Stanimirović, P. S., Li, S., & Cao, X. (2020). Time-varying minimum-cost portfolio insurance under transaction costs problem via beetle antennae search algorithm (bas). *Applied Mathematics and Computation*, *385*, Article 125453.

Kaucic, M. (2019). Equity portfolio management with cardinality constraints and risk parity control using multi-objective particle swarm optimization. *Computers & Operations Research*, *109*, 300–316.

Khan, A. H., Cao, X., Katsikis, V. N., Stanimirović, P., Brajević, I., Li, S., Kadry, S., & Nam, Y. (2019). Optimal portfolio management for engineering problems using nonconvex cardinality constraint: A computing perspective. *IEEE Access*, *8*, 57437–57450.

Khan, A. H., Cao, X., & Li, S. (2020). *Management and intelligent decision-making in complex systems: an optimization-driven approach*. Springer Nature.

Khan, A. T., Cao, X., Li, S., Hu, B., & Katsikis, V. N. (2023). Quantum beetle antennae search: a novel technique for the constrained portfolio optimization problem. *Science China. Information Sciences*, *64*(5), 1–14.

Khan, A. T., Cao, X., Li, S., Katsikis, V. N., Brajevic, I., & Stanimirovic, P. S. (2022). Fraud detection in publicly traded us firms using beetle antennae search: A machine learning approach. *Expert Systems with Applications*, *191*, 116–148.

Khan, A. T., Cao, X., Li, Z., & Li, S. (2021). Enhanced beetle antennae search with zeroing neural network for online solution of constrained optimization. *Neurocomputing*, *447*, 294–306.

Kim, D., & Francis, J. C. (2013). *Modern portfolio theory: Foundations, analysis, and new developments*. John Wiley & Sons.

Kremer, P. J., Lee, S., Bogdan, M., & Paterlini, S. (2020). Sparse portfolio selection via the sorted ℓ_1-norm. *Journal of Banking & Finance*, *110*, 105–137.

Li, S., He, J., Li, Y., & Rafique, M. U. (2017). Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(2), 415–426.

Li, Z., Liao, B., Xu, F., & Guo, D. (2020). A new repetitive motion planning scheme with noise suppression capability for redundant robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(12), 5244–5254.

Li, S., Zhang, Y., & Jin, L. (2018). Kinematic control of redundant manipulators using neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2243–2254.

Li, J., Zhang, Y., & Mao, M. (2021). Continuous and discrete zeroing neural network for different-level dynamic linear system with robot manipulator control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(11), 4633–4642.

Liao, B., Zhang, Y., & Jin, L. (2016). Taylor $o(h^3)$ discretization of znn models for dynamic equality-constrained quadratic programming with application to manipulators. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(2), 225–237.

Ma, Z., & Guo, D. (2020). Discrete-time recurrent neural network for solving bound-constrained time-varying underdetermined linear system. *IEEE Transactions on Industrial Informatics*, *17*(6), 3869–3878.

Markowitz, H. M. (1991). Foundations of portfolio theory. *The Journal of Finance*, *46*(2), 469–477.

Medvedeva, M. A., Katsikis, V. N., Mourtas, S. D., & Simos, T. E. (2021). Randomized time-varying knapsack problems via binary beetle antennae search algorithm: emphasis on applications in portfolio insurance. *Mathematical Methods in the Applied Sciences*, *44*(2), 202–212.

Miettinen, K. (2012). *Nonlinear multiobjective optimization*. Springer Science & Business Media.

Mohammed, A. M., & Li, S. (2015). Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms. *IEEE Transactions on Cybernetics*, *46*(7), 1538–1550.

Omisore, I., Yusuf, M., & Christopher, N. (2011). The modern portfolio theory as an investment decision tool. *Journal of Accounting and Taxation*, *4*(2), 19–28.

Qi, Y., Jin, L., Luo, X., & Zhou, M. (2022). Recurrent neural dynamics models for perturbed nonstationary quadratic programs: A control-theoretical perspective. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(3), 1216–1227.

Raei, R., & Alibeiki, H. (2010). Portfolio optimization using particle swarm optimization method. *Financial Research Journal*, *12*(29), 121–149.

Rebentrost, P., Gupt, B., & Bromley, T. R. (2018). Quantum computational finance: Monte carlo pricing of financial derivatives. *Physical Review A*, *98*(2), 302–321.

Ruszczynski, A. (2006). *Nonlinear optimization*. Princeton, NJ: Princeton University Press.

Sharma, S., Islam, N., Singh, G., & Dhir, A. (2022). Why do retail customers adopt artificial intelligence (ai) based autonomous decision-making systems. *IEEE Transactions on Engineering Management*, *3*(115–121), 1–16.

Shi, Y., Pan, Z., Li, J., Li, B., & Sun, X. (2022). Recurrent neural dynamics for handling linear equation system with rank-deficient coefficient and disturbance existence. *Journal of the Franklin Institute*, *359*(7), 3090–3102.

Shi, Y., & Zhang, Y. (2020). New discrete-time models of zeroing neural network solving systems of time-variant linear and nonlinear inequalities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(2), 565–576.

Tran, D. T., Iosifidis, A., Kanniainen, J., & Gabbouj, M. (2018). Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(5), 1407–1418.

Troiano, L., Villa, E. M., & Loia, V. (2018). Replicating a trading strategy by means of LSTM for financial industry applications. *IEEE Transactions on Industrial Informatics*, *14*(7), 3226–3234.

Tsantekidis, A., Passalis, N., Toufa, A.-S., Saitas-Zarkias, K., Chairistanidis, S., & Tefas, A. (2020). Price trailing for financial trading using deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(7), 2837–2846.

Xia, Y. (2004). Further results on global convergence and stability of globally projected dynamical systems. *Journal of Optimization Theory and Applications*, *122*(3), 627–649.

Xiao, L., He, Y., & Liao, B. (2022). A parameter-changing zeroing neural network for solving linear equations with superior fixed-time convergence. *Expert Systems with Applications*, *208*, Article 118086.

Xiao, L., Tao, J., Dai, J., Wang, Y., Jia, L., & He, Y. (2021). A parameter-changing and complex-valued zeroing neural-network for finding solution of time-varying complex linear matrix equations in finite time. *IEEE Transactions on Industrial Informatics*, *17*(10), 6634–6643.

Yaman, I., & Dalkilic, T. (2021). A hybrid approach to cardinality constraint portfolio selection problem based on nonlinear neural network and genetic algorithm. *Expert Systems with Applications*, *169*, 401–517.

Zhang, Y., & Ge, S. S. (2005). Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Transactions on Neural Networks*, *16*(6), 1477–1490.

Zhang, Z., Kong, L., Zheng, L., Zhang, P., Qu, X., Liao, B., & Yu, Z. (2020). Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying qm and qp problems and applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(12), 5106–5118.

Zhang, Y., Li, S., & Liu, X. (2018). Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(12), 6227–6241.

Zhang, Y., Li, S., Weng, J., & Liao, B. (2022). GNN model for time-varying matrix inversion with robust finite-time convergence. *IEEE Transactions on Neural Networks and Learning Systems*, *15*(1), 21–32.

Zhang, W., Liu, Y., & Xu, W. (2012). A possibilistic mean-semivariance-entropy model for multi-period portfolio selection with transaction costs. *European Journal of Operational Research*, *222*(2), 341–349.

Zhang, Y., Yang, M., Huang, H., Chen, J., & Li, Z. (2021). Unified solution of different-kind future matrix equations using new nine-instant discretization formula and zeroing neural dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–11.

Zhang, Z., Zheng, L., Yang, H., & Qu, X. (2019). Design and analysis of a novel integral recurrent neural network for solving time-varying sylvester equation. *IEEE Transactions on Cybernetics*, *PP*(99), 1–15.

Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications*, *38*(8), 10161–10169.