IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 June 2023, accepted 20 July 2023, date of publication 24 July 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298562

RESEARCH ARTICLE

Mercury: A Deep Reinforcement Learning-Based Investment Portfolio Strategy for Risk-Return Balance

ZENG-LIANG BAI, YA-NING ZHAO, ZHI-GANG ZHOU, WEN-QIN LI, YANG-YANG GAO, YING TANG, LONG-ZHENG DAI, AND YI-YOU DONG

School of Information, Shanxi University of Finance and Economics, Taiyuan 030006, China

Corresponding author: Zhi-Gang Zhou (zzgisgod@sina.com)

This work was supported in part by the National Science Foundation of China (NSF) under Grant 61902226, in part by the Philosophy and Social Science Planning Project of Shanxi Province under Grant 2022YY097, in part by the Key Project of "New Infrastructure and Informationization" for Higher Education by China Education Technology Association under Grant XJJ202205013, in part by the Fundamental Research Program of Shanxi Province under Grant 202203021221218, in part by the Youth Scientific Research Foundation of Shanxi University of Finance and Economics under Grant QN-2019021, and in part by the Scientific and Technology Innovation Programs of Higher Education Institutions in Shanxi under Grant 2019L0478.

ABSTRACT Stock portfolio is a hard issue in the Fintech field due to the diversity of data characteristics and the dynamic complexity of the market. Despite advances in deep learning that have made great progress in the complex and highly stochastic portfolio problem, the existing research still faces significant limitations. They either consider only investment returns or simply use some macro-market data to guide their models against risk. The preferred direction of the market greatly affects the choice of stock. And in practice, investors are more inclined to portfolios with low correlation between assets because of the ripple relationships between related things. In this paper, we propose a novel framework, called Mercury, which views stock screening as a reinforcement learning process. In particular, to enhance the ability to perceive changes in the market and generate higher returns, our framework models the sensitivity of the market preferences and learns dynamic temporal and spatial dependency patterns between assets from historical trading data. Additionally, the framework employs reinforcement learning to screen the overall low-correlation portfolio, which can better improve the ability to withstand investment risks while guaranteeing returns. The daily dataset of China's A-share market is used as the research sample to verify the effectiveness and robustness of Mercury, and our framework has strong generalization ability, which can be easily generalized to other trading procedures.

INDEX TERMS Deep reinforcement learning, risk-return balanced portfolio strategy, market preferences, low-correlation assets.

I. INTRODUCTION

A stock portfolio is a selection of stocks made according to specific rules and principles aimed at reducing investment risks. In brief, there are two primary reasons for constructing a stock portfolio: to diversify investment risks and to maximize investment returns. Presently, many studies utilize machine learning or deep learning techniques for predicting asset trends and subsequently selecting the top-performing assets to form an excellent portfolio. However, the majority of existing research [1], [2], [3], [4], [5] solely focus on investment returns or simply includes certain macro-market data as a risk indicator.

Historical data can only capture relevant features of short-term returns [6], [7], [8], [9], whereas qualitative information can provide a more complete picture of the underlying factors driving long-term trends [10]. Some analyses show that investment decision-makers rely more on qualitative information such as news, events, and even announcements when making decisions. Incorporating financial text data into the investment decision-making process can provide a more comprehensive and accurate understanding of the market [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

By modeling long-term trends in stocks, financial text data can provide a more robust basis for making investment decisions.

The stock market is inherently risky, and the level of risk is typically positively correlated with potential profits. However, investors generally aim to construct a portfolio that offers a balance between risks and returns, with lower risks and relatively high returns being the ideal scenario. One way to achieve this is by investing in different asset classes that are mutually independent or have low correlation, which can help to reduce overall portfolio risk [12], [13]. Intuitively, it can be viewed as a game between potential risk and reward in a framework similar to reinforcement learning (RL) to obtain a non-optimal but satisfactory strategy.

In this work, we formulate stock screening as a reinforcement learning process (Sec. III-A). The goal is to identify the optimal portfolio that balances return and risk. To facilitate the subsequent research, we integrate the core modules into the policy network (Sec. IV-B and IV-C): Firstly, we learn evolutionary features and correlations among stocks in a data-driven manner. Then, we incorporate market public sentiment indicators to model the long-term trend of stocks, and comprehensively evaluate the performance of individual stocks. Finally, we select portfolios with a lower overall association based on a set of well-performing stocks. Due to the discrete and non-differentiable nature of market fluctuations and trading mechanisms, we use policy gradient to jointly optimize targets in an end-to-end manner (Sec. IV-D). The main contributions of this paper are summarized as follows:

1) We simultaneously extract spatiotemporal features of trading data and capture inter-stock relations using hypergraph attention mechanisms. Modeling the long-term trend of the stock market by referring to the public sentiment indicator in natural language processing, and comprehensively evaluating the performance of the stock with multi-granularity.

2) We propose a novel ensemble framework Mercury that incorporates reinforcement learning into the stock screening process to generate a suitable portfolio that effectively withstands risk while guaranteeing returns.

3) All the modules are seamlessly integrated and jointly trained. Through the experiment on the real-world stock of China's A-share market, we demonstrate the applicability and effectiveness of Mercury in the quantitative portfolio with 50 stocks within 3305 trading days.

II. RELATED WORK

A. QUANTITATIVE PORTFOLIO

Quantitative investing is an investment strategy that uses mathematical and computer technology to guide investment decisions. Traditional quantitative methods mainly include statistical analysis, regression analysis, and time series analysis [14], [15], [16]. The core of these methods is to statistically analyze historical data to find patterns, trends and predict future market trends based on them. Existing mean regression strategies do not fully consider the noise and outliers of trading data. Given, Because of this problem, references [17] added a robust L1-median estimator to mean regression and proposed a robust median regression online portfolio selection strategy. However, these methods cannot accurately predict market changes, especially when facing complex and nonlinear markets. In recent years, with the rapid development of artificial intelligence technology, traditional quantitative methods are no longer the only choice.

Machine learning and deep learning can help us deal with these complex market data and improve the accuracy of predictions [7], [18], [19], [20], [21], [22]. Lim et al. [23] proposed deep momentum networks by combining trading rules based on deep learning with time series momentum strategies. Agrawal et al. [24] developed an Evolutionary Deep Learning Model (EDLM) that utilizes stock technical indicators to identify the prices of stock trends. Ding et al. [25] use a deep convolutional neural network to model both short-term and long-term effects of events on the movement of stock prices. Wang et al. [26] proposed an improved self-attention encoder, utilizing adaptive pattern interactions supported by temporal representations at different granularity, and constructed a data-driven adjacency graph to reveal the implicit similarity of volatility across different stocks.

Due to the development of reinforcement learning technology, the model combines deep neural networks with reinforcement learning for strategy transactions [27], [28], [29], [30], [31]. Under the reinforcement learning framework, EIIE [32] considers the weight of portfolio in network training and combines CNN, RNN, and LSTM three neural networks to achieve respectively. RAT [33] leverages transformer architecture to capture complex price sequence patterns of assets and price relationships between multiple assets for portfolio selection. However, the above methods do not involve additional market factors. DeepTrader [2] takes into account the interconnections and interactions between stocks and incorporates market conditions that work together to produce a risk-return balanced portfolio. While the methods mentioned above are effective in analyzing the correlation between assets and their sequential features, they may not be fully applicable to real-world investment scenarios where investors typically prefer a portfolio of assets with low correlation. Portfolio diversification is a commonly used strategy to minimize risk and maximize return by investing in a mix of assets that have a low correlation with each other.

B. CONVOLUTIONAL RECURRENT NEURAL NETWORK

CRNN is a general term for a series of convolutional neural networks (CNNs) combined with recurrent neural networks (RNNs) and derives from the graphic and text recognition task [34]. In a CRNN, the CNN layers are responsible for extracting features from the input data, while the RNN layers are responsible for processing the sequential information and capturing temporal dependencies. The CNN layers typically generate a fixed-length feature vector for each input sample, which is then fed into the RNN layers. The RNN layers

process the feature vectors sequentially and maintain a hidden state that captures the context of the previous inputs. By combining these two types of layers, a CRNN can effectively capture both the spatial and temporal information in the input data, making it useful for a wide range of applications, and in this case, spatio-temporal modeling for financial data analysis. ConvLSTM [35] is one of the pioneering works with a convolutional structure. Since then, other advanced spatio-temporal modeling networks have also been proposed, such as PredRNN [36], PredRNN++ [37] series, and E3D-LSTM [38]. In this paper, we select one of the pioneers and concise ConvLSTM as the module to extract features. The transformation between the states is shown below:

$$
\begin{aligned}
\mathbf{i}_t &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x}_t; \mathbf{W}_{\text{xi}}\right) + \text{Conv}\left(\mathbf{h}_{t-1}; \mathbf{W}_{\text{hi}}\right) + \mathbf{b}_{\text{i}}\right), \\
\mathbf{f}_t &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x}_t; \mathbf{W}_{\text{xf}}\right) + \text{Conv}\left(\mathbf{h}_{t-1}; \mathbf{W}_{\text{hf}}\right) + \mathbf{b}_{\text{f}}\right), \\
\mathbf{o}_t &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x}_t; \mathbf{W}_{\text{xo}}\right) + \text{Conv}\left(\mathbf{h}_{t-1}; \mathbf{W}_{\text{ho}}\right) + \mathbf{b}_{\text{o}}\right), \\
\mathbf{g}_t &= \text{Tanh}\left(\text{Conv}\left(\mathbf{x}_t; \mathbf{W}_{\text{xg}}\right) + \text{Conv}\left(\mathbf{h}_{t-1}; \mathbf{W}_{\text{hg}}\right) + \mathbf{b}_{\text{g}}\right), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \text{Tanh}\left(\mathbf{c}_t\right),
\end{aligned}
$$

where $\mathbf{W}_*$, $\mathbf{b}_*$ is the learnable weight and bias, $\odot$ is the Hadamard product.

III. PRELIMINARY

A. PROBLEM DEFINITION

Portfolio management is a sequential decision-making process that naturally fits into the framework of a Markov Decision Process (MDP) $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$. Herein, $\mathcal{S} = \{s_t\}$ is the set of states abstracting stock sequences during exploration, and $\mathbf{A} = \{a_t\}$ is the set of actions, which adds a stock to the current stock sequences at each step t. When action $a_t \in \mathbf{A}$ is executed, $s_{t-1} \in \mathbf{S}$ changes according to the transition distribution $s_t \sim \mathcal{T}(s_t \mid s_{t-1}, a_t)$. Next, the agent receives a reward $r_t = \mathbf{R}(s_{t-1}, a_t, s_t)$. Its goal is to learn a policy function, which action $a_t \in \mathbf{A}$ should be performed under the state s to maximize cumulative returns. Such as, the trajectory $(s_0, a_1, s_1, \cdots, a_t, s_t)$ naturally describes the formation of a portfolio, where the reward $r_t = \mathbf{R}(s_0, a_1, s_1, \cdots, a_t, s_t)$ reflects the returns under this portfolio. Here we elaborate on the foregoing key elements as follows.

1) STATE SPACE

At step t, the state s_t indicates the set of selected stocks that have a low correlation and perform well, where the initial state $s_0 = \emptyset$.

2) ACTION SPACE

Observing the state s_{t-1}, the available action space $\mathbf{A}_t$ is the complement of s_{t-1}, formally $\mathbf{A}_t = \mathbf{S} \setminus s_{t-1}$. The RL agent picks up a suitable stock from $\mathbf{A}_t$ to join in the previous selection s_{t-1}.

3) STATE TRANSITION DISTRIBUTION

Having made the action a_t at step t, the transition of the state s_t is merging a_t into the previous state s_{t-1}: $s_t = s_{t-1} \cup a_t$.

TABLE 1. Notations in the paper.

Notation	Description
N	total number of stocks
T	total number of trading days
D	the features dimension of each trading day for each stock
X_i^t	feature of stock i on trading day t
y_i^t	price rising rate of the stock i on trading day t
p_i^t	closing price of the stock i on trading day t
$\mathcal{S}, \mathrm{A}, \mathcal{T}, \mathrm{R}$	state, action, transition, and reward space
H	the dimension of hidden features
h_i	the hidden feature of the stock i
v_i, e_j	node v_i and its hyperedge e_j of hypergraph
$\hat{a}_{ij}$	attention coefficient of e_j to the stock node v_i
H_d	the dimension of projected feature space
$\mathbb{N}_i$	the neighborhood set of stock i
K	the number of attention heads
Z	stocks' representations
L	number of stocks in the action set
G	number of stocks in the portfolio
e^m	word vector of stock bar comment
h_t^m	sequential representation of input e^m at step t
M	the extent of the market's preference for stocks
r_i^t	rate of return for stock i on holding period t

4) REWARD DESIGN

We consider two factors in the reward design: Portfolio returns with the price rising rate and portfolio risks with the negative maximum drawdown as the reward function. It will be detailed later in Section IV-D.

B. TRADING PROCEDURE

The trading program in the $T+1$ market is more complex compared to the $T+0$ market. In the $T+1$ market, the delivery of stocks needs to be completed on the second trading day after buying or selling, which requires more time and procedures for settlement. In contrast, in the $T+0$ market, the delivery of stocks can be completed on the same trading day, and the trading process is relatively simple. Our strategy can easily generalize to the $T+0$ market.

At the end of the t-1 holding period, traders hold Q_0^{t-1} cash and $\boldsymbol{\beta}_{\mathbf{t-1}} = \{b_{t-1,1}, b_{t-1,2}, \cdots, b_{t-1,G}\}$ volume of stocks. The trader will finish the t period according to the following steps: 1) sell all stocks and wait for fund settlement; 2) meet with the remaining funds Q_0^{t-1} after receiving the cash; 3) reallocate funds based on new portfolio ratio ω and portfolio weight ρ for the next purchase.

IV. METHODOLOGY

A. FRAMEWORK OVERVIEW

Investment decisions rely on precise stock selection. Portfolio theory shows that diversified portfolios with low inter-asset correlation can effectively reduce individual risks [39]. Furthermore, in his seminal paper [40], Sharpe emphasizes the importance of optimizing portfolios to maximize risk-adjusted returns and achieve a more efficient risk-return tradeoff. Constructing well-diversified investment portfolios

can help investors reduce individual risks while obtaining better overall returns.

In the portfolio, the decision-maker needs to select a suitable asset portfolio to achieve specific investment goals. This process can be viewed as a series of decisions made in a sequence of time steps. Therefrom, we adopt a reinforcement learning framework to select candidate stocks with low correlation while guaranteeing returns.

Formally, given the candidate set $\mathbf{S} = \{s_1, \cdots, s_N\}$ of N stocks, on any trading day $t \in T$, each stock i contains a feature sequence $\mathbf{X}_i^t = \left[x_1^t, \cdots, x_D^t\right] \in \mathbb{R}^{T\times D}$, where D is the feature dimension. As shown in Figure 1, the input signals of 1) the Representation Learning and 2) Market Preference Sensitivity Modeling are the stock feature sequence $\mathbf{X}$ and the word vector of stock bar comment $\mathbf{e}^m$, respectively. The performance of the stocks is quantified as scores Y, which is used to select a set of actions $\mathbf{Z}$. Then, the portfolio is determined with the covariance of $\mathbf{Z}$, yielding the state $\mathbf{S}_G$. 3) The portfolio with low correlation is selected based on the action set. Next, the quantitative score $\mathbf{Y}_G$ of the state $\mathbf{S}_G$ is mapped to the portfolio weight and portfolio ratio. Specifically, the Action-Set $\mathbf{Z} = \{Z_1, \cdots, Z_L\}$ is obtained through the policy network, denoted as $\mathbf{Y}_L = \mathrm{PN}(\mathbf{X}, \mathbf{e}^m).\mathbf{Y}_L = \{y_1 > y_2 > \cdots > y_L\}$ and $L \leq N$. To obtain the $\mathbf{S}_G$ via $\mathbf{S}_G = \mathrm{LC}(\mathbf{Z})$. Subsequently, output the portfolio weight of G stocks and the proportion of funds according to $\mathbf{Y}_G$. Formulated as: $\rho, \omega = \mathrm{Generate}(\mathbf{Y}_G)$. Finally, we optimize the training goal using policy gradient.

B. POLICY NETWORK

Our policy network first learns the characteristic representation for each stock and utilizes the hypergraph attention (HGAT) network for embedding the neighbor information of the stocks. Considering the long-term impact of the market, model the sensitivity of the market and then predict the preference for each stock. Subsequently, the performance of all stocks is assessed based on stock representation and market factors. Take the well-performing stocks as an action set.

1) REPRESENTATION LEARNING OF STOCK CANDIDATES SET

The ConvLSTM layer is responsible for learning the spatiotemporal representation of the trading data, capturing its dynamics and spatial relationships. And the HGAT mechanism further learns the dependency relationships and importance among features. By combining the ConvLSTM layer and the HGAT mechanism, a more comprehensive feature representation can be obtained. The model can acquire richer feature representations that better reflect the interdependence and dynamic changes among stocks.

Specifically, we use the ConvLSTM [35] layer to handle spatial-temporal relations in long-range sequences. The ConvLSTM structure can not only establish the LSTM-like temporal relationship but also have a spatial feature extraction capability similar to the CNN. After conducting the ConvLSTM operation, we denote the output of this layer by $\mathbf{h} \in R^{N\times T\times H}$, where H is the dimension of hidden features,

$$\mathbf{h} = \mathrm{ConvLSTM}(\mathbf{X}). \tag{1}$$

We then use the hypergraph attention (HGAT) network to capture the inter-stock relationships. More specifically, the covariance matrix of the input $\mathbf{X} \in \mathbb{R}^{N\times T\times D}$ is calculated and the industry relationship of stocks is considered to jointly construct a hypergraph. This enables HGAT to learn dynamic and higher-order dependencies among assets. For each node v_i and its hyperedge e_j, we compute an attention coefficient $\hat{a}$ using the stock's temporal feature $\mathbf{h}_i$ and the aggregated hyperedge features $\mathbf{h}_j$, indicating the importance of the corresponding relationship e_j to the stock v_i. Each entry is further normalized via softmax to obtain $\hat{a}_{ij}$:

$$\hat{a}_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{f\in N_i}\exp\left(\mathrm{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_f]\right)\right)}, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{H\times H_d}$, $\mathbf{a} \in \mathbb{R}^{2*H_d\times 1}$ are the learnable parameter matrix, H_d is the dimension of projected feature space. $\mathbb{N}_i$ is the neighborhood set of stock i. Add the multi-head attention mechanism to stabilize the learning process and enhance the node representation. That is, K-independent attention heads are applied to compute the hidden states. Afterward, the final layer output is represented by the concatenation of all attention heads:

$$\mathbf{Z}_i = \|_{k=1}^{K}\sigma\left(\sum\nolimits_{j\in N_i}\hat{a}_{ij}^k\mathbf{W}^k\mathbf{h}_j\right), \tag{3}$$

where $\|$ is the concatenation operator and σ is an activation function. Thereby all stocks' representations are formed as $\mathbf{Z} = [Z_1; Z_2; \cdots; Z_N] \in \mathbb{R}^{N\times T\times KH_d}$.

2) MARKET PREFERENCE SENSITIVITY (MPS) MODELING

Given, Because of the highly stochastic and abruptness of stock data, investment strategies consistent with market tone will be more stable. Obtain market sentiment by modeling related stock bar comments, and predict the investment tendency of the market. Specifically, we crawled the stock bar comment data of the related stocks of Eastmoney.com and preprocessed the data such as cleaning and word segmentation. The text information is processed as the word vector $\mathbf{e}^m$ by Word2Vec, and uses a Long Short-Term Memory network to recursively extract the sequential representation of input $\mathbf{e}^{\mathbf{m}}$:

$$\mathbf{h}_t^m = \mathrm{LSTM}\left(\mathbf{h}_{t-1}^m, \mathbf{e}^m\right), t \in [1, T], \tag{4}$$

where $\mathbf{h}_t^m$ denotes the hidden state encoded by LSTM at step t. The last hidden state $\mathbf{H}_T$ can be seen as a global representation of the input signal.

3) TEMPORAL ATTENTION MECHANISM

Since information fever tends to decay over time, earlier information may not be effectively modeled over a long periods span. To model these characteristics, we adopt the

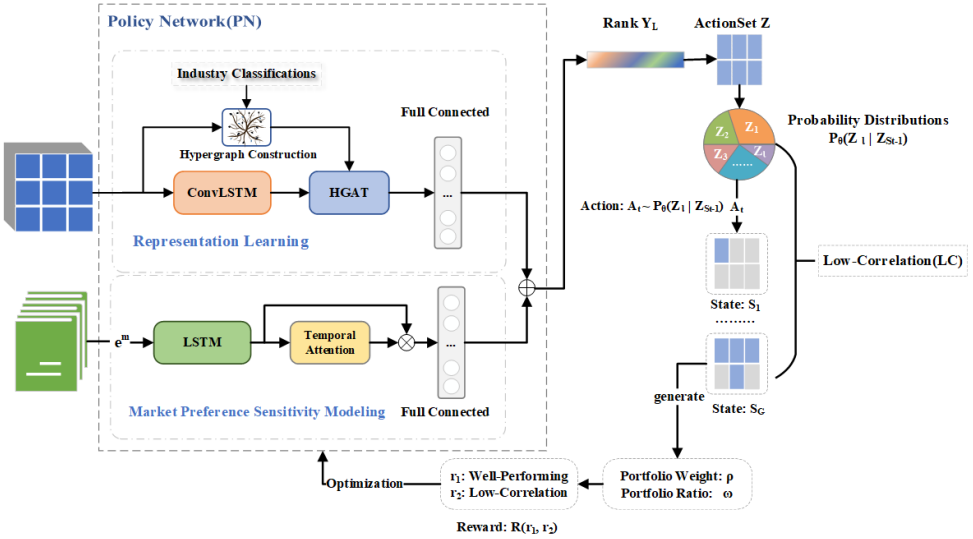

FIGURE 1. Mercury framework.

temporal attention mechanism to model the nonlinear relationship adaptively, and the attention weights are calculated as:

$$\alpha_t^m = \text{Softmax}\left(\mathbf{U}^T \tanh\left(\mathbf{W}_1^m \left[\mathbf{h}_t^m; \mathbf{H}_T\right]\right)\right), \tag{5}$$

where $\mathbf{U}$, $\mathbf{W}_*^m$ are parameters to learn, and b is the bias vector. The hidden state is further represented as the market's preference $\mathbf{M} \in R^{N\times 1}$ for each stock,

$$\mathbf{M} = \mathbf{W}^m(\sum_{t=1}^{T} \alpha_t^m * \mathbf{h}_t^m) + \mathbf{b}^m. \tag{6}$$

4) GETTING A WELL-PERFORMING ACTION SET

$\mathbf{Z}$ and $\mathbf{M}$ were added up to obtain the evaluation scores, where η is a parameter to balance two parts.

More specifically, we first sort all stocks in descending order based on score. Then we select the top L (L is greater than the number of invested stocks) stocks as a well-performing candidate set $\mathbf{Z} = [Z_1; Z_2; \cdots; Z_L] \in \mathbb{R}^{L\times T\times KH_d}$.

$$\text{score} = \text{Sigmoid}\left((\mathbf{W}\cdot\mathbf{Z} + \mathbf{b}) + \eta\mathbf{M}\right). \tag{7}$$

C. OVERALL LOW CORRELATION PORTFOLIO

1) SELECTION OF ACTION WITH OVERALL LOW CORRELATION (LC)

Having established the representation of action candidates, we aim to select one action from the space and perform it. Instead of trying candidates exhaustively, the policy network learns the importance of taking an action $a_t = \mathbf{Z}_l$ to the current state $s_{t-1} = \mathbf{Z}_{s_{t-1}}$:

$$\mathbf{C}_{l\in[1,\cdots,L]} = \text{Sigmoid}\left(\text{COV}\left(\left[\mathbf{Z}_l \parallel \mathbf{Z}_{s_{t-1}}\right]\right)\right), \tag{8}$$

where $\mathbf{Z}_l$, $\mathbf{Z}_{s_{t-1}}$ is the representation of the selected stock and current state, respectively. COV $(\cdot)$ denote the covariance operation of $\mathbf{Z}_l$ and $\mathbf{Z}_{s_{t-1}}$. Then we clamp this value into the range [0, 1] by Sigmoid, and values are directly proportional to the correlation. In other words, $\mathbf{C}_l$ indicates the correlation between the newly-added stock and the previously-selected stocks.

Thereafter, we apply an inverse proportional function (IPF) overall action candidates $\mathbf{A}_t$ to convert $\mathbf{C}_l$ into the probability distribution. The intuition is that actions with large covariance should have a smaller probability to be selected, ensuring a low correlation between stocks to balance risk.

$$\text{P}_\theta\left(\mathbf{Z}_l \mid \mathbf{Z}_{s_{t-1}}\right) = \text{Softmax}\left(\text{IPF}_{\mathbf{A}_t}\left(\mathbf{C}_l\right)\right). \tag{9}$$

Eventually, a low correlation sequence of G stocks is obtained, with G being the number of stocks given to the investment in advance.

2) PORTFOLIO GENERATING NETWORK (PGN)

PGN consists of two parts, respectively portfolio weight and portfolio ratio. When the prospect of the portfolio is good, give a larger proportion of investment, otherwise reduce the investment amount appropriately.

3) PORTFOLIO WEIGHT

After obtaining the portfolio state $\mathbf{S}_G$, we utilize the softmax function to transfer the previous evaluation score to portfolio weight ρ.

$$\rho = \begin{cases} \dfrac{\exp\left(\text{score}_j\right)}{\sum\limits_{i\in S_G}\exp\left(\text{score}_i\right)}, & j \in \mathbf{S}_G \\ 0, & \text{others.} \end{cases}$$

Portfolio Ratio: To better adapt to market conditions, a variable δ is set to dynamically adjust the proportion of the investment amount. Specifically, map the evaluation scores of the stocks to the range [1, 0] through an aggregation function F that consists of a linear layer and a sigmoid

function, namely $\delta = \text{Sigmoid}(\mathbf{W}_S \cdot \text{score} + \mathbf{b}_S)$. Given a threshold $\lambda = 0.5$, the portfolio ratio ω is divided into two levels:

$$\omega = \begin{cases} \lambda + \delta/2, & \text{if } \lambda \leq \delta \\ \lambda - \delta/2, & \text{others.} \end{cases}$$

D. OPTIMIZATION VIA POLICY GRADIENT

We use policy gradient to optimize the investment policy in an end-to-end manner. The reward consists of two parts: 1) select a well-performing candidate set based on historical data and market indicators, and 2) obtain the stock portfolio with low correlation on this basis.

Training Goal: The rate of return for holding period t is $r_i^t = Y_i^t \cdot \nu_\theta - 1$, where $\nu_\theta := \frac{\exp(\text{score}_i(\theta))}{\sum_{n=1}^{N} \exp(\text{score}_n(\theta))}$, $\text{score}_i(\theta)$ is the evaluation score of *ith* stock. $Y_i^t = \frac{p_i^t}{p_i^{t-1}}$ is the price rising rate and p_i is the closing price of the stock i. Given the initial investment amount Q_0, the cumulative wealth of a trajectory τ is $Q_{|\tau|} = Q_0 \prod_{\varepsilon=0}^{|\tau|} (r^t + 1) = Q_0 \prod_{\varepsilon=0}^{|\tau|} \mathcal{Y}_\varepsilon \cdot \nu_\theta$. In this way, the optimization goal is to maximize the log-accumulated wealth of a given trajectory.

Under the premise of ensuring wealth, we employ the negative maximum drawdown (MDD) rate as the reward function R_ε, to effectively measure the risk of the stock state s_t. In summary, the training goal is to maximize "(10)":

$$\begin{aligned} \mathcal{L}(\theta) = \sum\nolimits_{\tau \sim (\nu_\theta, \mathrm{P}_\theta)} \Bigg(& \sum\nolimits_{\varepsilon=0}^{|\tau|} \log\left(Y_\varepsilon \nabla \nu_\theta\right) \\ & + \gamma \sum\nolimits_{\varepsilon=0}^{|\tau|} R_\varepsilon \nabla log\left(\mathrm{P}_\theta\left(\mathbf{Z}_l \mid \mathbf{Z}_{s_{t-1}}\right)\right) \Bigg), \\ \theta \leftarrow \theta + \mu \mathcal{L}(\theta), & \end{aligned} \tag{10}$$

where $\boldsymbol{\gamma}$ is a parameter to balance the two goals. These two goals are trained simultaneously and updated $\boldsymbol{\theta}$ by gradient ascent with a learning rate $\boldsymbol{\mu}$.

V. EXPERIMENTS

To comprehensively evaluate the performance of Mercury, we conduct extensive and targeted experiments to answer the following questions: **Q1**: How are the portfolio profitability and risk performance generated by Mercury? **Q2**: How much do the core modules (such as MPS and LC) in Mercury contribute to the overall framework? **Q3**: How does Mercury personalize the portfolio for different users? **Q4**: What is the impact of different sizes of action sets L and the number of assets G on the portfolio?

A. EXPERIMENTAL SETUP

1) DATASETS

We have collected 50 representative stocks from various industries in the Chinese A-share market, A-50 for short, including trading data and its related stock bar information. As policies or traders change over time, the distribution structure of trading data may also change accordingly, and the use of too-long trading data is counterproductive. So, the time range of our data is from Jun. 2005 to Dec. 2018, with the period from Jun. 2005 to Dec. 2012 used as the training/validation set and the rest as the test set.

2) DETAILS

Mercury is implemented with PyTorch. We collect daily data on all stocks from the *Tushare* interface, including normalized opening-high-low-closing prices, trading volumes, and amounts. We follow [2] and generate data samples along the trade length by setting the window size to 7 days. For the ConvLSTM in presentation learning, the number of layers is set as 3, and the dimensions of hidden layers are 64, 64, and 32 respectively. In market preference sensitivity modeling, the attention heads of temporal attention mechanism K is 2. The batch size is 15 and the learning rate is 1e-06.

3) BASELINES

Mercury is compared with several baselines including:

- EIIE_LSTM [32] and DeepTrader (DT) [2]: two developed RL-based methods.
- Time Series Momentum (TSM) [6]: a classic momentum strategy.
- Robust Median Reversion (RMR) [17]: a newly reported reversion strategy.
- Relation-Aware Transformer (RAT) [33]: a novel attention-based method.
- Mercury/MPS: a model without the market preference sensitivity modeling structure, which means that the short-term returns of stocks are considered only.
- Mercury/LC: a model without the low correlation structure, which means that when selecting stock actions, it does not consider the overall low correlation (LC), and only selects the stocks with good performance.

4) METRICS

We use five indicators to evaluate our model, which can be roughly divided into three categories:

- Annualized Rate of Return (ARR) as a profit indicator.
- Annualized Volatility (AVL) and Maximum Drawdown (MDD) as risk indicators.
- Annualized Sharpe ratio (SR) and Calmar ratio (CR) are used as risk-profit indicators.

Among them, for risk indicators (AVL and MDD), the lower the better, and the higher the better for other indicators.

B. EXPERIMENTAL COMPARISON AND ANALYSIS

1) RESULTS 1 PERFORMANCE IN PROFITABILITY AND RISK

For **Q1**. Overall, the performance of Mercury is much better than other baselines. The goal of the portfolio is to achieve long-term returns. We can see that the overall performance of Mercury, Mercury/MPS, and Mercury/LC are better than other models, as shown in Table 2.

Figure 2 shows the portfolio wealth comparison of Mercury and the baselines. TSM and EIIE_LSTM have smaller investment returns, but their maximum drawdown is low and their performance is relatively flat. Conversely, RAT and

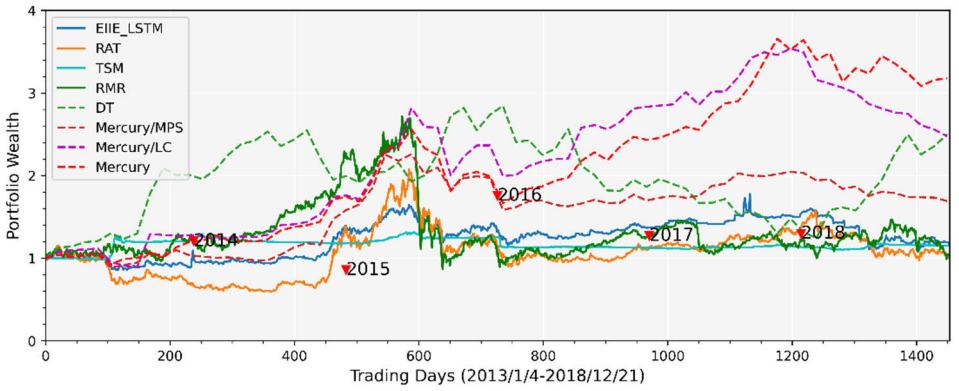

FIGURE 2. The portfolio wealth on A-50.

TABLE 2. Performance comparisons on different models.

Model	ARR	AVL	MDD	SR	CR
EIIE_LSTM	0.050	0.315	0.155	0.156	0.325
RAT	0.064	0.345	0.569	0.183	0.112
TSM	0.024	0.180	0.166	0.133	0.145
RMR	0.077	0.377	0.682	0.206	0.114
DT	0.091	0.275	0.463	0.292	0.174
Mercury/MPS	0.103	**0.230**	**0.334**	0.450	0.310
Mercury/LC	0.117	0.257	0.372	0.454	0.315
Mercury	**0.125**	0.245	0.338	**0.741**	**0.538**

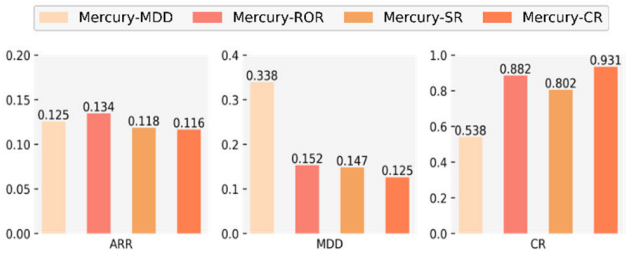

FIGURE 3. The performance of different reward functions.

RMR showed large fluctuations. The curve of RAT and RMR is interesting, moving relative to before 2015 and gradually overlapping after 2015. During the 2015 bull market, most portfolio returns rose to varying degrees and then fell. Among them, the performance of DT is different, with higher returns in the early stage (2013-2015) and an obvious decline when the general trend is good (early stage of 2015).

Among our three models, Mercury and Mercury/LC outperformed Mercury/MPS. It can be seen that before 2016, the trend of the three dashed lines is relatively consistent, and after 2016, Mercury/MPS begins to move away from the first two models. Compared with other models, our model is more robust in long-term profitability and risk.

2) RESULTS 2: ABLATION STUDY

For **Q2.** We perform ablation experiments with two simplified versions of Mercury. **Mercury/MPS** outperforms Mercury in terms of risk, this may be due to its lower ARR. In the stable and slow bull period (2016/6/30-2018/6/29), it can be found from Figure 2 that Mercury/MPS is relatively stable in this interval, with no obvious upward trend, and even a hint of fall. While Mercury and Mercury/LC in this interval performance are better. Verify the effectiveness of the MPS insight into the market environment.

All metrics of Mercury in Table 2 are better than **Mercury/LC**. However, the portfolio wealth curve of Mercury/LC in Figure 2 is comparable to Mercury overall, and even the trend is higher than Mercury before 2018. This is because in a better market, regardless of the low correlation between assets, a good stock will also lift the related stocks. A portfolio is a long-term process, and investors want relatively stable long-term returns. The trend in Figure 2 also indicates that Mercury/LC began to go downhill after experiencing a peak, with Mercury steadily rising.

3) RESULTS 3: PERSONALIZED PORTFOLIO

For **Q3.** To achieve personalized service, we investigate the effects of different choices of the reward function R_ε on the experimental results in the selection of low correlation action. Examples include rate of return (ROR), Sharpe ratio (SR), and Calmar ratio (CR). We use the form of the Mercury-ROR to indicate that the reward function used for the Mercury is the ROR.

In Figure 3, We choose the profit indicator ARR, risk indicator MDD, and risk-profit indicator CR to evaluate the performance of the reward function. Interestingly, MDD and CR of the other three reward functions have obvious changes compared with Mercury-MDD. Among them, Mercury-CR has the best risk resistance and good profit. Mercury-SR has a slightly higher ARR than Mercury-CR, but its MDD and CR performed less than Mercury-CR. Mercury-ROR has high ARR and low MDD, which is different from our original cognition, and the investment with high returns should often be

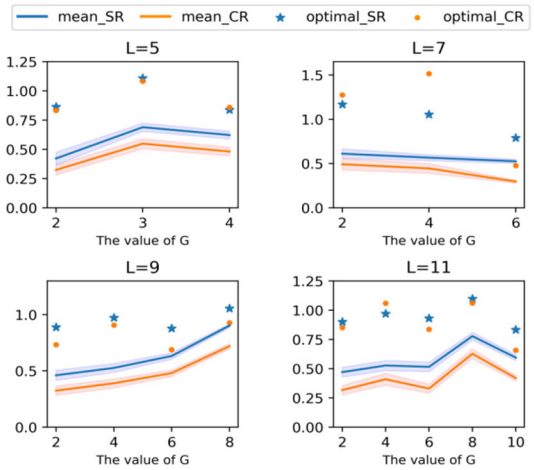

FIGURE 4. The performance of different values of L and G.

accompanied by high risk. This may be due to the bull market situation, by reducing the correlation between stocks, making portfolio risk resistance capacity better, are more likely to increase earnings. MDD is generally used for the worst-case scenario possible after a constant investment. As a reward function, Mercury-MDD reflects the worst performance of the portfolio in history, that is, our strategy can maintain an acceptable result for investors in a poor environment.

In summary, on the premise of resisting risks, Mercury-CR and Mercury-SR tend to provide cost-effective portfolios, Mercury-ROR has higher portfolio returns but may be volatile, and Mercury-MDD's portfolio tends to accumulate steadily.

C. RESULTS 4: EFFECT OF THE VALUES OF L AND G ON THE PORTFOLIO

For **Q4.** In the Mercury-MDD framework, we analyze the performance of the portfolio when L and G take different values. As shown in Figure 4, we calculate the mean values of the portfolio's SR and CR to reflect the stability and generalization ability of the strategy in terms of risk-return. Of course, considering the mean alone does not fully reflect its performance. Because we are usually more concerned with the optimal performance of investment strategy in practical applications. For a more comprehensive analysis and comparison, we visualized the mean and optimal values under different L and G portfolios.

It can be found that when $L = 5$, the gap between the mean value and the optimal value of SR and CR is small. When $G = 3$, both SR and CR are higher, indicating that both the returns and risks of the portfolio perform relatively well and have good investment value. At $L = 7$, the overall trend of SR and CR decreases with increasing G. Moreover, the distance between SR and CR is widening, indicating that the portfolio is more volatile. At $L = 9$, although the overall SR and CR are not high, the mean_SR and mean_CR are steadily rising. For $L = 11$, the portfolio performs best at $G = 8$, but it can be found that its SR and CR are not very different from those when $L = 5$, $G = 3$. This suggests that the two portfolios perform roughly equally well.

As a whole, a larger action set L can provide more options for constructing the portfolio and may offer greater diversification potential, but it can also increase the complexity of the optimization problem and make it more difficult to find an optimal solution. For example, in the case of $L = 11$. On the other hand, a smaller action set L can simplify the problem but may limit the potential for diversification and may result in a less optimal portfolio.

Similarly, the number of assets G in the portfolio can also have a significant impact on the portfolio's performance. A larger number of assets can provide greater diversification and may reduce the portfolio's risk, for example, when $L = 9$, the value scenario of G, but it can also increase the complexity of the portfolio construction problem and may result in higher transaction costs. On the other hand, a smaller number of assets can simplify the portfolio construction problem, but it may also limit the potential for diversification and may result in a portfolio that is more susceptible to risk.

In general, the optimal size of the action set L and the number of assets G will depend on the specific investment strategy being employed, and the goals of the portfolio. Experiments show that large L and G are detrimental to the performance of our strategy.

VI. CONCLUSION

We proposed the Mercury in the framework of reinforcement learning. In this paper, we considered the correlations between stocks and market conditions and combined historical data and market preferences of individual stocks to learn about assets that fit both tonalities. Stock screening was integrated into the learning process of reinforcement learning, and market preferences were also learned to jointly optimize the investment strategy with a low correlation between assets but acceptable returns. On this basis, we set the core modules into a policy network to easily replace them with more advanced models, which is conducive to our subsequent research work. Through the experiment in a real A-share stock market, to verify the effectiveness of the Mercury. And a simple modification of the trading procedure can easily generalize to the $T + 0$ market.

REFERENCES

[1] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, 2021, pp. 497–504.

[2] Z. Wang, B. Huang, S. Tu, K. Zhang, and L. Xu, "DeepTrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, 2021, pp. 643–650.

[3] S.-H. Huang, Y.-H. Miao, and Y.-T. Hsiao, "Novel deep reinforcement algorithm with adaptive sampling strategy for continuous portfolio optimization," *IEEE Access*, vol. 9, pp. 77371–77385, 2021, doi: 10.1109/ACCESS.2021.3082186.

[4] B. A. Luthfianti, D. Saepudin, and A. F. Ihsan, "Portfolio allocation of stocks in index LQ45 using deep reinforcement learning," in *Proc. 10th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Bandung, Indonesia, Aug. 2022, pp. 205–210, doi: 10.1109/ICOICT55009.2022.9914892.

[5] N. Darapaneni, A. Basu, S. Savla, R. Gururajan, N. Saquib, S. Singhavi, A. Kale, P. Bid, and A. R. Paduri, "Automated portfolio rebalancing using Q-learning," in *Proc. 11th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, New York, NY, USA, Oct. 2020, pp. 0596–0602, doi: 10.1109/UEMCON51285.2020.9298035.

[6] B. Lim, S. Zohren, and S. Roberts, "Enhancing time-series momentum strategies using deep neural networks," *J. Financial Data Sci.*, vol. 1, no. 4, pp. 19–38, Oct. 2019.

[7] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Appl. Soft Comput.*, vol. 70, pp. 525–538, Sep. 2018.

[8] L. Li, J. Wang, and X. Li, "Efficiency analysis of machine learning intelligent investment based on K-means algorithm," *IEEE Access*, vol. 8, pp. 147463–147470, 2020, doi: 10.1109/ACCESS.2020.3011366.

[9] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, Apr. 2019.

[10] J. Zheng, A. Xia, L. Shao, T. Wan, and Z. Qin, "Stock volatility prediction based on self-attention networks with social information," in *Proc. IEEE Conf. Comput. Intell. Financial Eng. Econ. (CIFEr)*, May 2019, pp. 1–7.

[11] T. Kabbani and E. Duman, "Deep reinforcement learning approach for trading automation in the stock market," *IEEE Access*, vol. 10, pp. 93564–93574, 2022, doi: 10.1109/ACCESS.2022.3203697.

[12] H. Zhu, S.-Y. Liu, P. Zhao, Y. Chen, and D. L. Lee, "Forecasting asset dependencies to reduce portfolio risk," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 4, 2022, pp. 4397–4404.

[13] N. David, "A deep learning ensemble to predict energy price direction and volatility on the asset financial market," *FUPRE J. Sci. Ind. Res.*, vol. 7, no. 1, pp. 71–81, 2023.

[14] C. L. Dunis, J. Laws, and P. Nam, *Applied Quantitative Methods for Trading and Investment*. Hoboken, NJ, USA: Wiley, 2003, pp. 239–291, doi: 10.1002/0470013265.

[15] G. Casella, S. Fienberg, and I. Olkin, "Springer texts in statistics," Design, Tech. Rep., 2006.

[16] F. Edition, "A guide to econometrics," in *Factors Affecting International Marriage Survival A Theoretical Approach*, 2017.

[17] D. Huang, J. Zhou, B. Li, S. Hoi, and S. Zhou, "Robust median reversion strategy for on-line portfolio selection," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2013, pp. 3–9.

[18] G. Wang, L. Cao, H. Zhao, Q. Liu, and E. Chen, "Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4418–4426.

[19] M. Yang, X. Zheng, Q. Liang, B. Han, and M. Zhu, "A smart trader for portfolio management based on normalizing flows," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, L. D. Raedt, Ed. Jul. 2022, pp. 4014–4021, doi: 10.24963/IJCAI.2022/557.

[20] D. Kisiel and D. Gorse, "Portfolio transformer for attention-based asset allocation," in *Proc. Artif. Intell. Soft Comput., 21st Int. Conf. (ICAISC)*. Zakopane, Poland: Springer, 2022, pp. 61–71.

[21] Y. Ma, W. Wang, and Q. Ma, "A novel prediction based portfolio optimization model using deep learning," *Comput. Ind. Eng.*, vol. 177, Mar. 2023, Art. no. 109023.

[22] J. Hao, F. He, F. Ma, S. Zhang, and X. Zhang, "Machine learning vs deep learning in stock market investment: An international evidence," *Ann. Operations Res.*, pp. 1–23, Mar. 2023.

[23] B. Lim, S. Zohren, and S. Roberts, "Enhancing time series momentum strategies using deep neural networks," *J. Financial Data Sci.*, vol. 1, no. 4, pp. 19–38, 2019.

[24] M. Agrawal, P. K. Shukla, R. Nair, A. Nayyar, and M. Masud, "Stock prediction based on technical indicators using deep learning model," *Comput., Mater. Continua*, vol. 70, no. 1, pp. 287–304, 2022.

[25] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–15.

[26] H. Wang, T. Wang, S. Li, J. Zheng, S. Guan, and W. Chen, "Adaptive long-short pattern transformer for stock investment selection," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 3970–3977.

[27] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, Mar. 2017.

[28] R. Wang, H. Wei, B. An, Z. Feng, and J. Yao, "Commission fee is not enough: A hierarchical reinforced framework for portfolio management," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, 2021, pp. 626–633.

[29] J. Wang, Y. Zhang, K. Tang, J. Wu, and Z. Xiong, "AlphaStock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1900–1908.

[30] Y.-H. Miao, Y.-T. Hsiao, and S.-H. Huang, "Portfolio management based on deep reinforcement learning with adaptive sampling," in *Proc. Int. Conf. Pervasive Artif. Intell. (ICPAI)*, Taipei, Taiwan, Dec. 2020, pp. 130–133, doi: 10.1109/ICPAI51961.2020.00031.

[31] U. Pigorsch and S. Schäfer, "High-dimensional stock portfolio trading with deep reinforcement learning," in *Proc. IEEE Symp. Comput. Intell. Financial Eng. Econ. (CIFEr)*, Helsinki, Finland, May 2022, pp. 1–8, doi: 10.1109/CIFER52523.2022.9776121.

[32] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," 2017, *arXiv:1706.10059*.

[33] K. Xu, Y. Zhang, D. Ye, P. Zhao, and M. Tan, "Relation-aware transformer for portfolio policy learning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4647–4653.

[34] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017, doi: 10.1109/TPAMI.2016.2646371.

[35] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[36] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[37] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5123–5132.

[38] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3D LSTM: A model for video prediction and beyond," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[39] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[40] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *J. Finance*, vol. 19, no. 3, pp. 425–442, Sep. 1964.

ZENG-LIANG BAI received the Ph.D. degree in optics from Shanxi University, Taiyuan, China, in 2017. He has been with the Shanxi University of Finance and Economics, since 2017. His research interests include machine learning, financial quantitative analysis, and information safety.

YA-NING ZHAO received the B.S. degree in computer science and technology from Xinzhou Normal University, Xinzhou, China, in 2021. She is currently pursuing the degree in computer application technology with the Shanxi University of Finance and Economics. Her research interests include reinforcement learning, deep learning, and financial quantitative analysis.

ZHI-GANG ZHOU received the Ph.D. degree in computer science from the Harbin Institute of Technology, in 2018. He is currently an Associate Professor of computer science with the Shanxi University of Finance and Economics, Taiyuan, China. With a background in data privacy protection. He has more than 20 publications in high-ranked journals and conferences, including IEEE INFOCOM and IEEE TRANSACTIONS ON CLOUD COMPUTING. He has made many research contributions to the area of big data analytics, security and privacy, the Internet of Things, and cloud computing. His research interests include financial quantitative analysis, security, and privacy, including multi-source data security fusion, trusted task offloading in mobile cloud computing, and federated machine learning.

WEN-QIN LI received the B.S. degree in computer science and technology from the Taiyuan University of Science and Technology, Shanxi, China, in 2012. She is currently pursuing the degree in computer application technology with the Shanxi University of Finance and Economics. Her research interests include reinforcement learning, deep learning, and financial quantitative analysis.

YANG-YANG GAO received the B.S. degree in network engineering from the Xi'an University of Technology, Xi'an, China, in 2020. He is currently pursuing the degree in computer application technology with the Shanxi University of Finance and Economics. His research interests include reinforcement learning, financial quantitative analysis, and federated learning.

YING TANG received the B.S. degree in software engineering from the Jiangxi University of Science and Technology, Jiangxi, China, in 2022. He is currently pursuing the degree in technology for computer applications with the Shanxi University Finance and Economics, Taiyuan. His research interests include financial quantitative analysis, deep learning, and reinforcement learning.

LONG-ZHENG DAI received the B.S. degree in computer science and technology from the North University of China, Taiyuan, China, in 2022. He is currently pursuing the degree in computer application technology with the Shanxi University of Finance and Economics. His research interests include reinforcement learning, financial quantitative analysis, and deep learning interpretability.

YI-YOU DONG received the B.S. degree in computer science and technology from the Beijing Institute of Petrochemical Technology, Beijing, China, in 2022. He is currently pursuing the degree in technology for computer applications with the Shanxi University of Finance and Economics, Taiyuan. His research interests include federated learning, financial quantitative analysis, and reinforcement learning.