**Operations Research and Fuzziology** 运筹与模糊学, 2025, 15(3), 128-139
Published Online June 2025 in Hans. https://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2025.153146

Hans汉斯

# 基于风险价值探索机制的PPO-DBN算法股票交易策略

**李冠男，张国凯***

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月26日；录用日期：2025年5月30日；发布日期：2025年6月9日

## 摘　要

**随着人工智能技术与算法的发展，其在金融交易市场决策中的应用越来越广泛。特别是使用深度强化学习方法模拟交易环境实现交易决策成为当前的研究热点。基于此，本文提出基于风险价值探索机制的PPO-DBN算法，将近端策略优化(Proximal Policy Optimization, PPO)算法结合深度信念网络(Deep Belief Net, DBN)，并在训练中使用基于风险价值的探索机制，使用当前市场的风险价值(Value at Risk, VaR)动态调整$\varepsilon$-greedy的探索率。并且为了更好掌握市场数据的变化情况，引入基于波动率驱动的自适应移动平均值(Adaptive Moving Average, AMA)来构造状态空间，根据市场波动率动态调整均线窗口，同时，使用日资产变化作为奖励函数进行算法训练。最后，将该算法应用在中国股票市场中的六组股票行情数据进行实验验证。实验结果表明，所提出的算法在夏普比率、收益率等多个评价指标上均有良好表现。**

## 关键词

**深度强化学习，交易决策，股票交易，自适应移动平均值**

# Stock Trading Strategy Based on RaV Exploration Mechanism with PPO-DBN Algorithm

**Guannan Li, Guokai Zhang***

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 26th, 2025; accepted: May 30th, 2025; published: Jun. 9th, 2025

*通讯作者。

文章引用: 李冠男, 张国凯. 基于风险价值探索机制的 PPO-DBN 算法股票交易策略[J]. 运筹与模糊学, 2025, 15(3): 128-139. DOI: 10.12677/orf.2025.153146

## Abstract

**With the development of artificial intelligence technologies and algorithms, their applications in decision-making in the financial trading market have become increasingly widespread. In particular, the use of deep reinforcement learning methods to simulate the trading environment and achieve trading decisions has become a current research hotspot. Based on this, this paper proposes the PPO-DBN algorithm based on the risk value exploration mechanism. It combines the Proximal Policy Optimization (PPO) algorithm with the Deep Belief Net (DBN), and uses the risk value-based exploration mechanism during the training process. The Value at Risk (VaR) of the current market is used to dynamically adjust the exploration rate of the $\varepsilon$-greedy algorithm. Moreover, in order to better grasp the changes in market data, an Adaptive Moving Average (AMA) driven by volatility is introduced to construct the state space. The moving average window is dynamically adjusted according to the market volatility. At the same time, the daily asset change is used as the reward function for algorithm training. Finally, this algorithm is applied to the market data of six groups of stocks in the Chinese stock market for experimental verification. The experimental results show that the proposed algorithm performs well in multiple evaluation indicators such as the Sharpe ratio and the rate of return.**

## Keywords

**Deep Reinforcement Learning, Trade Decision, Stock Trading, Adaptive Moving Average**



## 1. 引言

金融历史交易数据有不稳定、高噪声、高维度等特点[1]。并且市场价格受多种因素影响[2]，在这种情况下，人工方法难以在噪声干扰下充分挖掘数据的内在特征，最终导致交易决策中无法实现理想的收益效果。近年来，随着人工智能技术的发展，其在金融交易市场中的应用吸引了大量学者的关注，尤其是，机器学习和深度学习等技术在交易决策领域的应用，为投资者带来了可观的收益。因此，将人工智能技术融入金融交易，可以更加准确地获取市场行情数据的变化特征，从而帮助投资者降低风险并提高收益。

为此，本文采用深度强化学习构建基于风险价值探索机制的 PPO-DBN 算法，以解决股票市场数据波动大、维度高及特征难以学习的问题，从而在单支股票交易中实现投资回报最大化。该方法通过智能体对环境中状态的持续学习来计算最优交易决策，并使用强化学习将股票投资的整个过程视为一个马尔可夫决策过程，该过程由状态、动作、奖励、策略以及相应的值来表示[3]。在状态空间构造时添加自适应移动平均线，该值基于历史波动率来确定动态滑动窗口大小计算对应移动平均值得到，当波动率较大时减小滑动窗口大小，从而减小波动噪声干扰；当波动率较小时增加滑动窗口大小以减少滞后性。与此同时，使用 PPO [4]算法来训练智能体，将其与 DBN [5]相结合并添加风险价值计算以动态调整训练过程中的探索率以学习数据特征，通过风险价值计算得到的探索率会随市场的波动而变化，当风险价值越大时，探索率减小，从而达到在高风险状态下自动降低随机探索概率来限制单次交易最大损失的目的。并通过累计收益(Cumulative Return, CR)、年化收益率(Annualized Return, AR)、最大回撤(Maximum Drawdown,

MD)和夏普比率(Sharpe Ratio, SR)等指标来评估该策略的性能。

经过实验证明，所提出的方法在中国 A 股市场中表现良好。本文的主要贡献可以概括为以下三点：

(1) 提出基于风险价值探索机制的 PPO-DBN 算法，该算法在结合 PPO 和 DBN 网络的基础上，引入风险价值动态调整探索率，以限制单次交易的最大损失，实现降低投资风险的目的。

(2) 引入自适应移动平均指标来拓展状态空间维度，根据价格波动性自动调整平滑系数，以在趋势市中减少滞后性、在震荡市中降低噪音干扰，从而提高算法模型对市场趋势和变化的感知能力。

(3) 将提出的算法应用于中国股票市场的 A 股数据中，实验结果表明，该算法在夏普比率、收益率等多个评价指标中展现了显著的有效性。

## 2. 相关工作

在人工智能方法中，包括机器学习、深度学习及强化学习方法。在这一部分将分别介绍这三种方法在金融交易领域的应用。

Lin 等人引入了一种基于机器学习的 K 线图形态识别模型，该模型在考虑交易成本的前提下，也实现了盈利[6]。此外，Lotfi 等人发现，通过整合人工智能方法，如机器学习和深度学习算法，考虑金融序列中非理性主体的行为，可以改进对资产价格变化的预测[7]。这些研究突出了机器学习在交易中的潜力及其在各个方面优于传统方法的能力。

另外一些研究尝试了深度学习技术在股票交易中的应用。例如，Selvamuthu 等人使用人工神经网络在预测印度股票市场方面取得了很高的准确率。展现了其在做出准确交易决策方面的潜力[8]。Yang 等人提出了一个结合卷积神经网络和长短期记忆网络的框架，在预测股票价格走势方面优于其他模型[9]。Nabipour 等人发现，循环神经网络和长短期记忆网络是利用连续数据预测股票市场趋势的最佳模型，而深度学习方法在处理二元数据时表现更好[10]。Yang 等人提出了一种基于卷积神经网络的股票交易框架，通过考虑股票数据的异质性和突发性，强调了纳入深度学习模型以捕捉股票市场复杂模式的重要性[11]。Lin 等人展示了他们的 LSTMGA 模型在利用混合数据进行股票市场预测方面的优越性能。该模型证明了结合多种信息来源来预测股票价格的有效性[12]。Yu 等人开发了一个智能轻型股票交易系统，该系统集成了深度学习模型和技术分析方法。他们确定，卷积神经网络注意力双向长短期记忆网络模型在预测股票价格方面最为有效[13]。

也有部分学者将强化学习方法应用于金融交易市场提升投资回报。Santos 等人发现，纳入商品衍生品可改善投资组合表现，在不增加风险的情况下使回报率提高 12% [14]。Yang 等人提出了 TC-MAC 算法，该算法优于传统方法和其他强化学习算法[15]。Lin 等人开发了一个用于投资组合管理的深度强化学习框架，与既定策略相比表现出优越的性能[16]。总之，强化学习在交易中的应用在改善投资组合管理策略、增强投资者对资产相关性的理解以及开发自动化股票交易系统方面已显示出较好的成果。

## 3. 问题建模

本文提出的基于风险价值探索机制的 PPO-DBN 算法股票交易策略，通过添加动态探索机制，并使用自适应移动平均值来扩展状态空间，实现算法在面对不同风险时能灵活处理数据并调整探索概率的目的。

### 3.1. 决策算法 PPO-DBN

决策算法的框架结构如图 1 所示，主要由两个核心部分构成：策略网络 Actor 与价值网络 Critic。其整体处理过程如下：

首先完成网络参数的初始化，初始化之后将 Actor 网络参数复制给与之结构完全相同的 Actor_old 网

络，在算法训练过程中 Actor 网络作为训练网络，Actor_old 网络作为目标网络共同训练。完成参数复制后，会在训练数据集中选择 batch_size 大小的数据分别输入 Actor 网络、Actor_old 网络及 Critic 网络，三个网络经过计算后分别输出当前环境下的 Actor 网络动作概率、Actor_old 网络动作概率及状态价值。下一步使用计算得到的状态价值和 Actor_old 网络动作的对数概率计算优势函数 advantage 及 Critic 网络的训练损失 critic_loss，继续根据计算得到的优势函数 advantage 计算 Actor 网络的 actor_loss，进而使用 actor 网络计算得到动作概率计算分类分布计算得到交叉熵损失。最后，根据三部分损失计算训练网络的总损失从而更新网络参数。

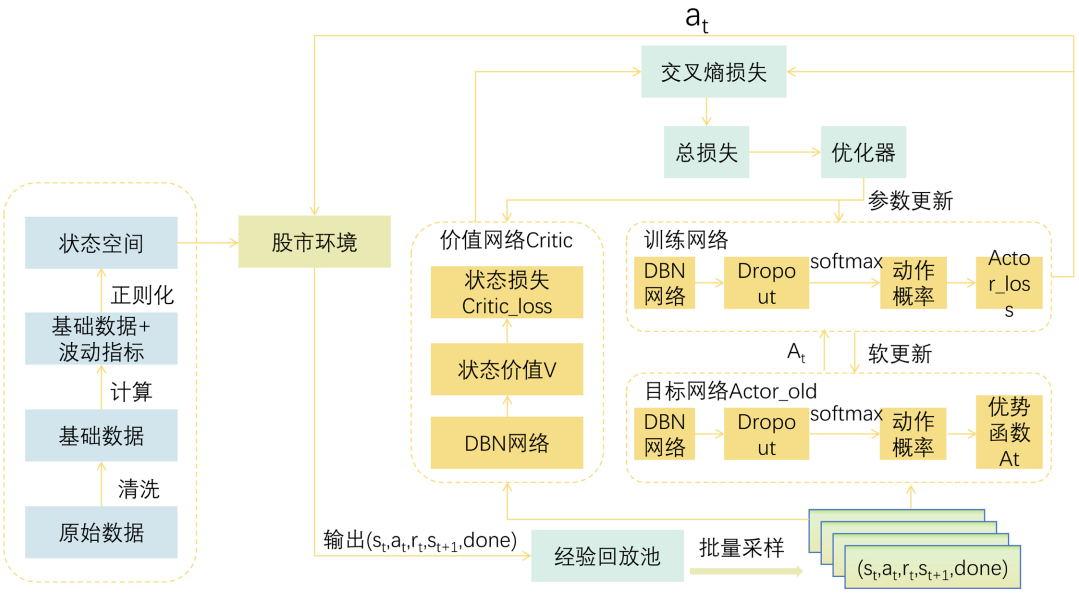


**Figure 1.** Flow chart of PPO-DBN algorithm processing
**图 1.** PPO-DBN 算法处理流程图

## 3.2. 环境定义

为了将相应的算法应用于股票交易问题，在本文中将状态定义为当前交易市场数据组成的向量，将动作定义为市价买入、市价卖出以及持有股票的决策。每次交易执行后的奖励定义为相邻两个状态之间资产的变化，具体如下：

### 3.2.1. 状态空间

由于设计的算法用于训练单支股票的交易策略，因此将状态空间定义为一个维度为 $6\times T_w$ 的向量，其中 $T_w$ 是每次训练的时间窗口，单个交易日的数据向量为 $\{O_t, C_t, H_t, L_t, V_t, A_t\}$，设定时间窗口内所有交易日的市场数据构成本次训练的状态空间并输入到智能体中进行学习和训练。

状态空间中每个属性的含义如下：$O_t$ 表示当日市场的开盘价，$C_t$ 是收盘价，$H_t$ 是最高价，$L_t$ 是最低价，$V_t$ 是当日成交量，$A_t$ 是预处理计算得到的当前时间步下计算得到的自使用移动平均值。状态空间中的所有值都是经过预处理和标准化后的值，取值范围均为[0, 1]。其中 $A_t$ 的计算方式如下：

$$A_t = \begin{cases} \dfrac{1}{t+1}\sum_{k=0}^{t} C_k, & t < \text{base\_window} \\ \dfrac{1}{w_t}\sum_{k=t-w_t+1}^{t} C_k, & t \geq \text{base\_window} \end{cases} \tag{1}$$

其中，$A_t$ 表示基于动态窗口 $w_t$ 内的自适应移动平均指标。$C_k$ 为时间窗口内第 $k$ 日的收盘价。base_window 为预定义的基准窗口大小。 $w_t$ 为动态滑动窗口大小，计算方式如下：

$$w_t = \text{clip}\left(\text{round}\left(\text{base\_window} \times \frac{\text{ref\_vol}}{\rho_t}\right), 5, 60\right) \tag{2}$$

上式中 ref_vol 为参考波动率，默认设置为 0.02， $\rho_t$ 为当前波动率，并且动态窗口 $w_t$ 被限制在[5, 60]之间以避免极端值。其中 $\rho_t$ 的计算方式如下：

$$\rho_t = \text{std}\left(R_{t-\text{vol\_window}+1}, R_{t-\text{vol\_window}+2}, \cdots, R_t\right) \tag{3}$$

在式(3)中 vol_window 表示滚动窗口，默认设置为 10， $R_i$ 表示指定时间窗口内第 $i$ 天的收益，其计算方式如下：

$$R_i = \frac{C_i - C_{i-1}}{C_{i-1}} \times 100\% \tag{4}$$

### 3.2.2. 动作空间

在本研究中，所有输入经过神经网络计算后输出为一个 1 × 3 的向量{0, 1, 2}，从 0 到 2 分别代表市价卖出、持有股票和市价买入动作。该向量再经过 Softmax 层计算得到一个由 3 个动作概率组成的向量，智能体在对应的概率向量中依据探索策略选择动作并执行相应交易。

### 3.2.3. 奖励函数

在本研究中，将奖励函数定义为相邻两个状态之间资产的变化，即从状态 $s_{t-1}$ 执行动作 $a_t$ 到下一个状态 $s_t$ 时资产的变化，可描述如下：

$$r\left(s_{t-1}, a_t, s_t\right) = \text{NAV}_t - \text{NAV}_{t-1} \tag{5}$$

式(5)中 $\text{NAV}_t$ 表示状态 $s_t$ 下的资产净值(Net Asset Value, NAV)，计算方式如下：

$$\text{NAV}_t = b_t + p_t \times C_t - c_t \tag{6}$$

上式中 $b_t$ 表示在时刻 $t$ 个人账户中的可用资金量， $p_t$ 是当前时刻持仓数量， $C_t$ 是当前状态下的收盘价， $c_t$ 表示本次交易产生的佣金。设定每次交易成本为每次交易金额的 0.1%，则 $c_t$ 计算方式如下：

$$c_t = 0.001 \times \left|p_t - p_{t-1}\right| \times C_t \tag{7}$$

### 3.2.4. 探索策略

在进行算法训练时，使用基于风险价值的动态 $\varepsilon$-greedy 策略来平衡探索性动作的选择。在计算风险价值前需先计算滑动窗口风险，需将收益率取负值转换为损失并存储在滑动窗口中：

$$L_t = -R_t, \ \mathcal{L} = \left\{L_{t-w+1}, L_{t-w+2}, \cdots, L_t\right\} \tag{8}$$

取损失序列的 $\alpha$-分位数计算风险价值：

$$\text{VaR}_\alpha = Q_\alpha\left(\mathcal{L}\right) \tag{9}$$

其中，$\alpha$ 为 VaR/CVaR 的置信水平，默认 0.95。对超过 VaR 的损失求均值，计算条件风险价值(Conditional Value at Risk, CVaR)：

$$\text{CVaR}_\alpha = \begin{cases} E\left[L \middle| L \geq \text{VaR}_\alpha\right], & |L| \geq 10 \\ 0.05, & |L| < 10 \end{cases} \tag{10}$$

滑动窗口风险计算完成后，即可使用自适应探索率机制计算动态探索率。计算时将输入风险限制在[0, 0.2]范围内：

$$\text{Risk}_t = \text{clip}\left(\text{CVaR}_\alpha, 0, 0.2\right) \tag{11}$$

完成之后进行指数衰减调整：

$$\varepsilon_t = \varepsilon_{\min} + \left(\varepsilon_{\max} - \varepsilon_{\min}\right) \times \text{e}^{-\beta \cdot \text{Risk}_t} \tag{12}$$

上式中 $\varepsilon_{\min}$ 为最小探索率，默认取 0.01， $\varepsilon_{\max}$ 为最大探索率，默认取 0.03， $\beta$ 为风险敏感系数，默认取 2.0。使用上述方法计算得到动态探索率之后，在训练过程中以计算得到的 $\varepsilon_t$ 为概率进行随机探索。

## 4. 实验数据与分析

### 4.1. 数据集

在本实验中，随机选取中国 A 股市场六支股票十年的历史行情数据，时间跨度为 2014 年 1 月 1 日至 2023 年 12 月 31 日。该数据集被划分为三个独立时段：2014 年 1 月 1 日至 2022 年 12 月 31 日的数据作为训练集；2023 年 1 月 1 日至 2023 年 12 月 31 日的数据则专门用于测试集。
(https://www.tushare.pro/webclient/)

### 4.2. 评价指标

累计收益率(CR)衡量交易策略在特定周期内产生的总收益或总亏损。计算公式如下：

$$\text{CR} = \frac{P_{\text{end}} - P_0}{P_0} \tag{13}$$

其中， $P_0$ 指投资周期初始时的资产净值； $P_{\text{end}}$ 指投资周期结束时的资产净值。

年化收益率(*AR*)指将投资周期的累计收益率转换为年度基准，便于不同周期策略的标准化比较。计算公式如下：

$$\text{AR} = \frac{\text{CR}}{W_t} \cdot 365 \cdot 100\% \tag{14}$$

式(14)中 cr 为投资周期内的累计收益率； $W_t$ 为投资周期的总天数。

最大回撤(*MD*)指通过历史数据量化策略可能产生的极端亏损，反映策略在最坏情况下的风险敞口。计算方式如下：

$$\text{MD} = \frac{\max\left(P_i - P_j\right)}{P_i} \tag{15}$$

上式中 $P_i$ 、 $P_j$ 满足 $j > i$ 的任意两日资产净值； $\max\left(P_i - P_j\right)$ 为统计周期内观测到的最大资产净值跌幅。

夏普比率(SR)用来衡量单位风险产生的超额收益，综合评估策略的收益﹣风险平衡能力。计算公式如下：

$$\text{SR} = \frac{\left(R_p - R_f\right)}{\sigma_p} \tag{16}$$

在 SR 的计算中， $R_p$ 为投资周期内的预期收益， $R_f$ 为无风险利率(基于中国人民银行最新公布的短期国债收益率，当前值为 2.08%)， $\sigma_p$ 为组合年化收益率的波动率(标准差)。

### 4.3. 实验结果分析

#### 4.3.1. 自适应移动平均值计算窗口探索

为避免极端值对自适应移动平均值计算的影响，在计算过程中需要定义计算 AMA 时滑动窗口的取值范围。在本实验中，进行六组不同滑动窗口取值范围的结果对比。取值范围与实验结果见表 1。

**Table 1.** Statistics of AMA backtesting results calculated by different sliding windows
**表 1.** 不同滑动窗口计算 AMA 回测结果统计

| | Window_size | [5, 50] | [5, 60] | [5, 80] | [10, 50] | [10, 60] |
|---|---|---|---|---|---|---|
| 000004.sz | CR | 19.11% | **44.17%** | 30.86% | 29.36% | 12.88% |
| | MD | 25.11% | 45.85% | 36.27% | 37.50% | **13.80%** |
| | AR | 29.30% | **85.10%** | 56.30% | 55.84% | 17.09% |
| | SR | 0.98 | **1.47** | 1.15 | 1.10 | 1.39 |
| 000030.sz | CR | 11.99% | **24.87%** | 8.22% | 15.47% | 8.19% |
| | MD | 13.98% | 25.03% | **9.17%** | 17.12% | 9.60% |
| | AR | 14.31% | **31.52%** | 9.75% | 18.77% | 9.66% |
| | SR | 1.27 | **1.44** | 1.00 | 1.31 | 1.08 |
| 000050.sz | CR | 16.58% | **16.95%** | 10.30% | 9.72% | 16.04% |
| | MD | 21.51% | 24.73% | **17.62%** | 25.88% | 24.18% |
| | AR | 21.72% | **22.41%** | 13.38% | 15.02% | 21.52% |
| | SR | **1.39** | 1.33 | 1.01 | 0.56 | 1.17 |
| 600007.sh | CR | 20.57% | **22.05%** | 9.41% | 13.82% | 10.15% |
| | MD | 22.52% | 23.33% | **12.42%** | 16.73% | 15.54% |
| | AR | 26.81% | **28.72%** | 11.60% | 17.85% | 13.01% |
| | SR | 1.10 | **1.16** | 0.77 | 0.84 | 0.68 |
| 600060.sh | CR | 9.10% | **34.69%** | 9.10% | 31.00% | 6.14% |
| | MD | 10.22% | 28.18% | 10.66% | 25.77% | **7.82%** |
| | AR | 10.83% | **46.33%** | 10.87% | 41.60% | 7.23% |
| | SR | 1.04 | **1.54** | 1.00 | 1.39 | 0.84 |
| 600066.sh | CR | 6.38% | **45.54%** | 5.95% | 2.19% | 17.28% |
| | MD | 9.29% | 40.08% | 9.39% | **2.59%** | 19.65% |
| | AR | 7.69% | **61.99%** | 7.31% | 2.54% | 21.53% |
| | SR | 0.73 | **1.89** | 0.59 | 0.28 | 1.28 |
| 平均值 | CR | 13.96% | **31.38%** | 12.31% | 16.93% | 11.78% |
| | MD | 17.11% | 31.22% | 15.92% | 20.93% | **15.10%** |
| | AR | 18.44% | **46.01%** | 18.20% | 25.27% | 15.00% |
| | SR | 1.09 | **1.47** | 0.92 | 0.91 | 1.07 |

由上表数据可知，在回测过程中，当计算 AMA 滑动窗口取值定义在[5, 60]时，算法性能整体表现最优，值得注意的是在最大回撤上表现不如其他取值，但可以看到在最大回撤增加约一倍的情况下累积收

益及年化收益增加近两倍，所以在后续训练中该参数取[5, 60]进行训练。

#### 4.3.2. 动态探索批次大小

在使用动态探索策略时，需根据前文中所计算的不同风险值确定探索空间来决定采样批量大小，不同的采样批量大小会对动态探索的结果产生影响。在本实验中根据动态风险的结果设置两种探索策略，当风险大于10%时使用$bs_{max}$大批量探索，提高稳定性；小于10%时使用$bs_{min}$小批量探索获取当前状态。针对探索批次大小设置以下四组取值进行实验，实验结果见表2。

**Table 2.** Statistics of backtesting results for different batch sizes
**表2.** 不同批次范围回测结果统计

| | $\{bs_{max}, bs_{min}\}$ | {16, 32} | {16, 64} | {16, 128} | {32, 64} |
|---|---|---|---|---|---|
| 000004.sz | CR | **44.17%** | 20.22% | 36.71% | 15.85% |
| | MD | 45.85% | 27.83% | 42.71% | **23.02%** |
| | AR | **85.10%** | 36.40% | 69.96% | 26.26% |
| | SR | **1.47** | 0.86 | 1.29 | 0.77 |
| 000030.sz | CR | **24.87%** | 6.86% | 6.37% | 8.25% |
| | MD | 25.03% | **7.84%** | 7.86% | 9.61% |
| | AR | **31.52%** | 8.05% | 7.47% | 9.74% |
| | SR | **1.44** | 1.01 | 0.96 | 1.08 |
| 000050.sz | CR | **16.95%** | 16.08% | 9.27% | 12.64% |
| | MD | **24.73%** | 25.06% | 25.70% | 25.84% |
| | AR | **22.41%** | 21.49% | 14.68% | 17.78% |
| | SR | **1.33** | 1.20 | 0.53 | 0.81 |
| 600007.sh | CR | **22.05%** | 11.85% | 2.31% | 3.89% |
| | MD | 23.33% | 15.62% | **2.87%** | 4.72% |
| | AR | **28.72%** | 15.28% | 2.65% | 4.51% |
| | SR | **1.16** | 0.76 | 0.24 | 0.60 |
| 600060.sh | CR | **34.69%** | 8.10% | 32.32% | 29.95% |
| | MD | 28.18% | **9.10%** | 26.66% | 25.78% |
| | AR | **46.33%** | 9.67% | 41,75% | 40.34% |
| | SR | 1.54 | 0.91 | **1.63** | 1.35 |
| 600066.sh | CR | **45.54%** | 17.77% | 40.31% | 12.58% |
| | MD | 40.08% | 21.40% | 37.56% | **16.35%** |
| | AR | **61.99%** | 23,40% | 54,50% | 15,65% |
| | SR | **1.89** | 1.01 | 1.73 | 1.00 |
| 平均值 | CR | **31.38%** | 13.48% | 21.22% | 13.86% |
| | MD | 31.22% | 17.81% | 23.89% | **17.55%** |
| | AR | **46.01%** | 17.75% | 31.84% | 17.44% |
| | SR | **1.47** | 0.96 | 1.06 | 0.94 |

由上表数据可见，在批次大小设置为{16, 32}时算法性能整体表现最佳。在后续实验中使用此参数进行训练。

#### 4.3.3. 对比实验

在本文中，选取 SSE50，SZI，Sandwich [17]，KRNN [17]，ALSTM [17]，AdaRNN [18]，Timemixer [19]，Itransformer [20]的公开方法进行对比，由于获取到的源码为批量训练，所以在进行对比时将本文实验得到的六组数据结果进行均值计算，使用均值与基线方法对比，详细对比数据见表 3，图 2 和图 3 分别展示所提出的算法与对比方法的累积收益与年化收益统计数据：

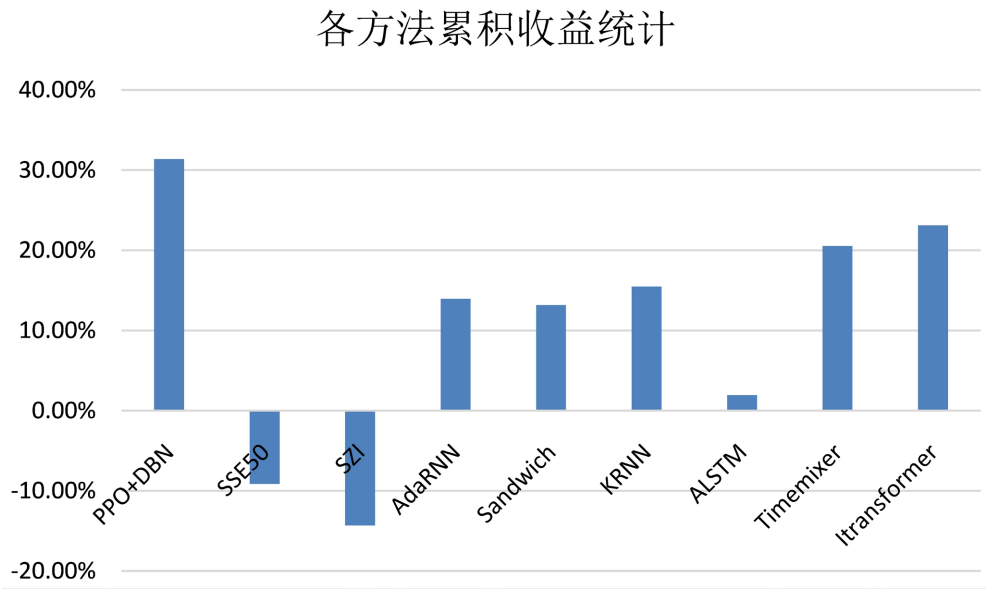


**Figure 2.** Statistics of the cumulative returns during the backtesting process of each method
**图 2.** 各方法回测过程中累积收益统计

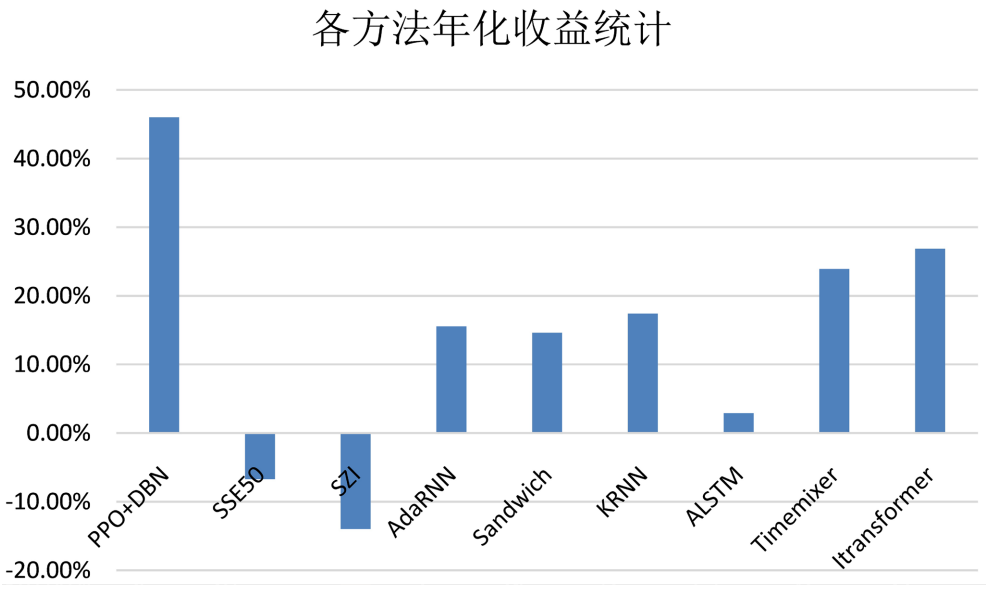


**Figure 3.** Statistics of the annualized returns during the backtesting process of each method
**图 3.** 各方法回测过程中年化收益统计

**Table 3.** Statistics in comparison with the baseline method
**表 3.** 与基线方法对比统计

| | CR | AR | MD | SR |
|---|---|---|---|---|
| PPO-DBN | **31.38%** | **46.01%** | 31.22% | **1.47** |
| SSE50 | −9.15% | −6.71% | 31.84% | −0.35 |
| SZI | −14.32% | −13.97% | 24.68% | −1.08 |
| Sandwich | 13.17% | 14.63% | 39.92% | 0.40 |
| KRNN | 15.48% | 17.41% | 34.71% | 0.41 |
| ALSTM | 1.94% | 2.91% | **18.94%** | 0.60 |
| AdaRNN | 13.95% | 15.56% | 29.36% | 0.39 |
| Timemixer | 20.53% | 23.91% | 22.03% | 1.20 |
| Itransformer | 23.12% | 26.86% | 20.21% | 1.32 |

由表中数据可见，所提出的算法在累积收益、年化收益、夏普比率三个指标均优于其他方法，但最大回撤与部分方法相比还有差距。这证明所提出算法在收益方面是有效的，但是收益带来了更大的资产波动。

### 4.3.4. 消融实验

**Table 4.** The results of the ablation experiments of VaRPPO-DBN on various datasets
**表 4.** VaRPPO-DBN 在各个数据集上消融实验结果

| | 模型 | CR | AR | MD | SR |
|---|---|---|---|---|---|
| 000004.SZ | VaRPPO-DBN | **44.17%** | **85.10%** | 45.85% | **1.47** |
| | w/o VaR | 28.18% | 52.32% | **35.84%** | 1.07 |
| | SMA | 34.60% | 65.52% | 38.08% | 1.26 |
| 000030.SZ | VaRPPO-DBN | **24.87%** | **31.52%** | 25.03% | **1.44** |
| | w/o VaR | 17.68% | 21.87% | **13.30%** | 1.26 |
| | SMA | 17.51% | 21.44% | 19.04% | 1.34 |
| 000050.SZ | VaRPPO-DBN | 16.95% | 22.41% | 24.73% | **1.33** |
| | w/o VaR | **20.53%** | **23.91%** | **22.03%** | 1.20 |
| | SMA | 11.69% | 17.07% | 25.87% | 0.70 |
| 600007.SH | VaRPPO-DBN | **22.01%** | **28.72%** | 23.33% | **1.16** |
| | w/o VaR | 17.73% | 22.99% | **20.34%** | 0.99 |
| | SMA | 20.39% | 26.66% | 22.72% | 1.08 |
| 600060.SH | VaRPPO-DBN | **34.69%** | **46.33%** | 28.18% | **1.54** |
| | w/o VaR | 25.16% | 33.17% | **23.03%** | 1.25 |
| | SMA | 28.20% | 38.05% | 24.16% | 1.28 |
| 600066.SH | VaRPPO-DBN | **45.54%** | **61.99%** | **40.08%** | **1.88** |
| | w/o VaR | 29.86% | 40.63% | 32.94% | 1.35 |
| | SMA | 18.99% | 26.13% | 25.30% | 0.95 |

为了进一步分析 VaRPPO-DBN 中所使用的 AMA 和基于风险价值的动态探索策略的有效性，本节对所提出方法进行了消融研究。w/o VaR 表示 VaRPPO-DBN 未使用基于风险价值的动态探索而在探索策略中 $\varepsilon$ 初始为 1 并使用固定折扣因子线性下降的版本；SMA 表示 VaRPPO-DBN 中使用简单移动平均线 SMA 代替 AMA 的版本；消融实验数据如表 4 所示。

## 5. 结论

本文提出了基于风险价值探索机制的 PPO-DBN 算法来进行股票交易决策的制定，该算法将 PPO 算法与 DBN 网络结合来处理股票交易市场的数据，使用 AMA 扩展状态空间，并添加基于风险因子的动态探索机制用于算法训练。并在中国股市 A 股市场中的六支股票进行验证，经过实验综合对比分析说明该算法在股票交易中对于收益提升有效。未来将进一步验证该算法在全球市场中的有效性，并结合网络实时数据、动态舆情等信息扩充算法训练的状态空间，使其能够更加准确地预测股价的走势从而帮助投资者获取更大的收益。

## 参考文献


[1] 杨胜刚, 卢向前. 行为金融、噪声交易与中国证券市场主体行为特征研究[J]. 湖南大学学报(社会科学版), 2002(1): 25-29.

[2] Kiboi, J. and Katuse, P. (2015) Nairobi Stock Exchange: A Regression of Factors Affecting Stock Prices. *Prime Journal of Social Science*, **4**, 1093-1098.

[3] van Otterlo, M. and Wiering, M. (2012) Reinforcement Learning and Markov Decision Processes. In: Wiering, M. and van Otterlo, M., Eds., *Reinforcement Learning*, Springer, 3-42. https://doi.org/10.1007/978-3-642-27645-3_1

[4] Sutton, R.S. and Barto, A.G. (2018) Reinforcement Learning: An Introduction. MIT Press.

[5] Hua, Y.M., Guo, J.H. and Zhao, H. (2015) Deep Belief Networks and Deep Learning. *Proceedings of* 2015 *International Conference on Intelligent Computing and Internet of Things*, Harbin, 17-18 January 2015, 1-4. https://doi.org/10.1109/icaiot.2015.7111524

[6] Lin, Y., Liu, S., Yang, H., Wu, H. and Jiang, B. (2021) Improving Stock Trading Decisions Based on Pattern Recognition Using Machine Learning Technology. *PLOS ONE*, **16**, e0255558. https://doi.org/10.1371/journal.pone.0255558

[7] Lotfi, I. and El Bouhadi, A. (2021) Artificial Intelligence Methods: Toward a New Decision Making Tool. *Applied Artificial Intelligence*, **36**, Article ID: 1992141. https://doi.org/10.1080/08839514.2021.1992141

[8] Selvamuthu, D., Kumar, V. and Mishra, A. (2019) Indian Stock Market Prediction Using Artificial Neural Networks on Tick Data. *Financial Innovation*, **5**, Article No. 16. https://doi.org/10.1186/s40854-019-0131-7

[9] Yang, C., Zhai, J. and Tao, G. (2020) Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory. *Mathematical Problems in Engineering*, **2020**, Article ID: 2746845. https://doi.org/10.1155/2020/2746845

[10] Nabipour, M., Nayyeri, P., Jabani, H., S., S. and Mosavi, A. (2020) Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms via Continuous and Binary Data; a Comparative Analysis. *IEEE Access*, **8**, 150199-150212. https://doi.org/10.1109/access.2020.3015966

[11] Yang, K., Zhang, G., Bi, C., Guan, Q., Xu, H. and Xu, S. (2023) Improving CNN-Based Stock Trading by Considering Data Heterogeneity and Burst. *International Journal on Cybernetics & Informatics*, **12**, 01-13. https://doi.org/10.5121/ijci.2023.120201

[12] Lin, Y., Lai, C. and Pai, P. (2022) Using Deep Learning Techniques in Forecasting Stock Markets by Hybrid Data with Multilingual Sentiment Analysis. *Electronics*, **11**, Article 3513. https://doi.org/10.3390/electronics11213513

[13] Yu, S., Yang, S. and Yoon, S. (2023) The Design of an Intelligent Lightweight Stock Trading System Using Deep Learning Models: Employing Technical Analysis Methods. *Systems*, **11**, Article 470. https://doi.org/10.3390/systems11090470

[14] Santos, G.C., Garruti, D., Barboza, F., de Souza, K.G., Domingos, J.C. and Veiga, A. (2023) Management of Investment Portfolios Employing Reinforcement Learning. *PeerJ Computer Science*, **9**, e1695. https://doi.org/10.7717/peerj-cs.1695

[15] Yang, S. (2023) Deep Reinforcement Learning for Portfolio Management. *Knowledge-Based Systems*, **278**, Article ID: 110905. https://doi.org/10.1016/j.knosys.2023.110905

[16] Lin, Y., Chen, C., Sang, C. and Huang, S. (2022) Multiagent-Based Deep Reinforcement Learning for Risk-Shifting Portfolio Management. *Applied Soft Computing*, **123**, Article ID: 108894. https://doi.org/10.1016/j.asoc.2022.108894

[17] Zhao, L., Kong, S. and Shen, Y. (2023) DoubleAdapt: A Meta-Learning Approach to Incremental Learning for Stock Trend Forecasting. *Proceedings of the* 29*th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, 6-10 August 2023, 3492-3503. https://doi.org/10.1145/3580305.3599315

[18] Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., *et al*. (2021) AdaRNN: Adaptive Learning and Forecasting of Time Series. *Proceedings of the* 30*th ACM International Conference on Information & Knowledge Management*, Queensland, 1-5 November 2021, 402-411. https://doi.org/10.1145/3459637.3482315

[19] Wang, S.Y., *et al*. (2024) TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. arXiv:2405.14616.

[20] Liu, Y., *et al*. (2023) iTransFormer: Inverted Transformers Are Effective for Time Series Forecasting. arXiv: 2310.06625.