

分类号_____

密级_____

U D C_____

编号_____

復旦大學

博 士 后 研 究 工 作 报 告

算法交易策略研究：基于机器学习、深度学习等方法的探索

陈隽德

工作完成日期 2021 年 12 月—2023 年 12 月

报告提交日期 2023 年 12 月

复 旦 大 学 (上 海)

2023 年 12 月

算法交易策略研究：基于机器学习、深度学习等方法的探索

Research on Algorithmic Trading Strategies: An Approach Based on
Machine Learning, Deep Learning, and Others

博 士 后 姓 名：陈隽德

流动站（一级学科）名称：应用经济学

专 业（二级学科）名称：金融学

指 导 教 师：牛晓健、贾西贝

研究工作起始时间：2021 年 12 月

研究工作期满时间：2023 年 12 月

复 旦 大 学（上 海）

2023 年 12 月

内容摘要

本文综合探讨了量化投资领域内的核心概念、机器学习算法的应用、以及针对特定市场环境开发的交易策略。在量化投资的基础框架中，我们重点介绍了投资组合管理和算法交易的基本理念，旨在通过数学模型和计算机技术最大化投资回报并控制风险。随后，本文深入分析了机器学习在量化投资中的多重作用，覆盖了传统机器学习、深度学习、强化学习及集成学习等算法，并探讨了它们在算法交易中的具体应用场景。

特别地，本文开发了一种基于 XGBoost 集成算法的高频算法交易策略，并详细说明了该策略在数据处理、模型训练及策略回测方面的实施细节。此外，针对国内 A 股市场的特性，本文提出了一套 T+0 交易策略，该策略利用深度强化学习和分钟级数据预测股价走势，结合算法交易技术生成交易信号，展示了在动态市场环境中寻求超额回报的可能性。

本文的研究不仅拓展了量化投资策略的理论与实践范围，而且通过结合最新的机器学习技术，展示了如何应对市场的复杂性和不确定性。未来研究将进一步探索新兴技术在量化投资中的应用，旨在为投资决策提供更加科学、高效的支持系统。

关键词：量化交易，算法交易，市场微观结构，机器学习，深度学习

ABSTRACT

This report comprehensively discusses the core concepts within the field of quantitative investment, the application of machine learning algorithms, and trading strategies developed for specific market conditions. In the fundamental framework of quantitative investment, we primarily introduce the basic ideas of portfolio management and algorithmic trading, aiming to maximize investment returns and control risks through mathematical models and computer technology. Subsequently, the report delves into the multifaceted roles of machine learning in quantitative investment, covering algorithms such as traditional machine learning, deep learning, reinforcement learning, and ensemble learning, and explores their specific applications in algorithmic trading.

Particularly, this study developed a high-frequency algorithmic trading strategy based on the XGBoost ensemble algorithm and detailed the implementation specifics in data processing, model training, and strategy backtesting. Moreover, tailored to the characteristics of China's A-share market, the report proposes a T+0 trading strategy that uses deep reinforcement learning and minute-level data to predict stock price trends and generate trading signals combined with algorithmic trading techniques, demonstrating the potential for seeking excess returns in a dynamic market environment.

The research not only expands the theoretical and practical scope of quantitative investment strategies but also showcases how to address the complexity and uncertainty of the market by integrating the latest machine learning technology. Future research will further explore the application of emerging technologies in quantitative investment, aiming to provide a more scientific and efficient support system for investment decisions.

KEY WORDS: quantitative trading, algorithmic trading, market micro-structure,

machine learning, deep learning

目 次

第一章 绪论	1
1.1 量化投资的基本概念	1
1.2 投资组合管理	3
1.2.1 理论基础	3
1.2.2 投资组合构建	6
1.2.3 评价体系	7
1.3 算法交易	7
1.3.1 市场微观结构	9
1.3.2 交易成本分析	12
1.3.3 传统算法交易策略	15
第二章 机器学习与算法交易	19
2.1 机器学习简介	19
2.1.1 传统机器学习	19
2.1.2 深度学习	25
2.1.3 强化学习	31
2.1.4 集成学习	35
2.2 机器学习在算法交易中的应用	36
2.2.1 订单簿模拟	36
2.2.2 价格预测	37
2.2.3 情绪分析	38
2.2.4 合规检测	38
第三章 基于机器学习的高频算法交易研究	41
3.1 研究背景	41
3.2 交易系统	41
3.3 研究方法	43
3.3.1 策略的定义	43
3.3.2 数据处理	43
3.3.3 模型构建	50
3.3.4 训练结果及分析	53
3.4 结论与讨论	55

第四章 基于强化学习的中国 A 股 T+0 策略研究	57
4.1 研究背景	57
4.2 研究方法	58
4.2.1 数据处理	58
4.2.2 模型构建	60
4.2.3 T+0 选股	64
4.3 交易结果及分析	65
4.4 结论与讨论	66
全文总结	69
附录 A 附表	71
参考文献	73

第一章 绪论

金融市场自诞生以来，受到技术、经济、政治和社会因素的共同影响，持续演变并不断发展。从传统交易方式到现代高度自动化的市场，投资行为逐渐演变成为一门关于投资的科学，发展出一类利用数学、统计学和定量分析等方法来制定投资策略的投资方式——量化投资。

在欧美等发达金融市场，量化投资已经有几十年的历史，其有效性经历了市场长期的检验。我国量化投资兴起于 2010 年左右，并在之后经历了快速的发展。根据中国证券投资基金业协会的统计，截止 2021 年末，已备案的量化基金共有 16850 只，规模达到了 1.08 万亿元¹。随着量化投资规模的不断扩大，对量化投资的研究需求也在不断增加。本章将简要介绍量化投资及其相关概念。

1.1 量化投资的基本概念

量化投资的概念十分宽泛。理论上，只要使用数学、统计学或其他定量分析方法制定的投资策略都可以归类为量化投资。学界通常将量化投资分析师（Quant）分为两种类型^[1]：P-Quant 和 Q-Quant。P-Quant 通常适用于买方市场。他们的主要角色是根据历史数据预测未来趋势，然后确定交易策略。Q-Quant 通常适用于金融机构的卖方。他们的主要任务是根据数学模型对金融衍生品进行定价。表1-1展示了它们的部分区别。本文所讨论的对象是买方量化，即 P-Quant。如无特殊说明，下文量化投资皆指买方量化（P-Quant）。

	Q-Quant	P-Quant
业务	卖方，如投资银行等	买方，如对冲基金等
测度	风险中性（risk neutral）	真实概率（real probability）
目标	推断现在	预测未来
工具	随机过程、偏微分方程	时间序列分析，贝叶斯统计，机器学习等

表 1-1 P-Quant 与 Q-Quant^[1]。

买方量化投资主要分为两个部分：投资组合管理（Portfolio Management, QM）和算法交易（Algorithmic Trading, TA）。这两部份共同覆盖了从投资组合构建到交易执行的全过程。图1-1展示了投资组合管理和算法交易之间的关系，其中，横

¹ 中国私募基金行业数据报告: <https://www.amac.org.cn/researchstatistics/report/zgsmjjhysjbg/>

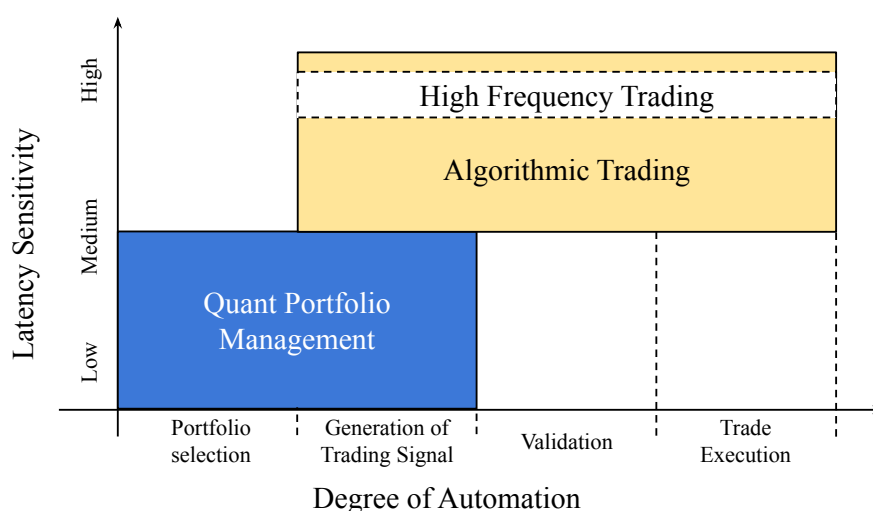


图 1-1 投资组合管理与算法交易、高频交易的关系^[2]。

坐标代表研究范围及其自动化程度，纵坐标表示时间延迟敏感度。可以看到，从投资组合构建到交易执行，自动化程度逐渐增加。

投资组合管理重点在于通过分析交易数据、基本面信息和市场情绪等来构建交易组合，目的是获得超过市场基准的超额收益。其研究范围通常包含组合构建及低频信号的生成，例如股票市场中的选股与择时。这一过程对延迟的敏感度较低，可以使用非常复杂的模型进行研究。

算法交易则主要基于对市场微观结构的理解，利用低延迟技术在短时间内迅速处理大量订单，从而在微小的市场波动中获取利润。其研究范围通常涵盖高频信号的生成、算法验证以及交易执行三个过程。算法交易对延迟敏感度和硬件技术的要求都比较高，需要高效交易系统的支持。

Hasbrouck et al. [3]将算法交易分为专有算法（Proprietary Algorithm, PA）和代理算法（Agency Algorithm, AA）两类。代理算法通过帮助交易者制定交易策略来节约交易成本。专有算法的目的则是直接从交易环境中获利，一般包括做市和统计套利两种类型，通常用于资产管理公司和对冲基金的自营产品。与代理算法相比，专有算法对速度延迟更加敏感，是狭义上的高频交易^[4]，目前，学术界和各国监管机构尚未对高频交易给出明确统一的定义^[5-9]。表1-2比较了代理算法与专有算法的部份相同点和不同点。

投资组合管理主要关注资产的中长期收益，而不太会涉及市场冲击、交易合规等交易层面的具体细节。相比之下，算法交易的目标是通过精心设计的交易策略，来减少交易的摩擦成本或通过统计套利等方法直接从交易中获益。需要注意的是，交易执行中的摩擦成本对投资组合的整体收益有显著影响。因此，一个高效的量

化投资策略应当将投资组合管理与算法交易有效结合起来。

代理算法	专有算法（高频交易）
预设计的交易决策，实时观察市场数据， 自动提交订单，自动订单管理，没有人为干预	
以基准价格为交易目标 最小化市场冲击 随着时间和市场的变化处理订单	通过买卖差价获利（做市、套利） 迅速处理撤单 持仓周期非常短，需要在日内平仓

表 1-2 代理算法与专有算法的共同特征与独有特征^[2]。

1.2 投资组合管理

在现代量化投资领域，多因子模型被广泛认为是构建投资组合的核心方法之一。本节将简要介绍多因子投资组合模型，包含多因子模型的理论基础，投资组合的构建过程及其评价体系。

1.2.1 理论基础

1.2.1.1 Markowitz 投资组合理论

Markowitz [10]提出了现代投资组合理论（Modern Portfolio Theory, MPT），认为资产的价格仅由收益的期望与方差决定。对某一资产，若其收益率 r_i 出现的概率为 p_i ，那么该资产的期望收益率 $E(r) = \sum_i p_i r_i$ ，其风险由方差描述， $\sigma_r^2 = \sum_i p_i [r_i - E(r)]^2$ 。如果对 N 个资产进行组合，则有，

$$E(r_p) = \sum_{i=1}^N w_i E(r_i), \quad \sigma_p^2 = \sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_i \sigma_j \rho_{ij}, \quad (1-1)$$

其中 w_i 为资产 i 的投资权重，且 $\sum_i w_i = 1$ ， ρ_{ij} 为资产 i 与 j 的关联系数。在 $E(r_p) - \sigma_p$ 图中，取 σ_p 对应的最大 $E(r_p)$ 所形成的一条曲线，称为投资组合的有效边界。根据有效边界理论，组合资产比组合中任一资产拥有更高的期望收益，当投资者的风险等级确定后，有效边界上的任意一点即为该风险下的最优投资组合。

Markowitz 模型第一次定量的给出了构建最优投资组合的方法，但也极大的简化了市场逻辑。Markowitz 假设：（1）所有投资者都是风险厌恶者；（2）投资者进行的是单期且静态投资；（3）交易不存在摩擦成本；（4）市场不包含无风险资产

等。此外，Markowitz 模型中的资产收益率 r_i 及其出现概率 p_i 是对历史数据的统计，造成了实际应用时缺乏稳定性。

1.2.1.2 资本资产定价模型

Sharpe [11], Treynor [12], Lintner [13], Mossin [14] 在 Markowitz 模型的基础上提出了资本资产定价模型 (Capital Asset Pricing Model, CAPM)。对组合中的任一资产 i ，其收益率可以表示为 $r_i = \alpha_i + \beta_i r_m + \epsilon_i$ ，其中 α_i 是一个常数， r_m 为市场收益率， ϵ_i 是一个独立的随机变量，其期望 $E(\epsilon_i) = 0$ 。系数 β_i 称为灵敏度系数或因子载荷、因子暴露，描述该证券在市场风险中所占的比重，

$$\beta_i = \frac{Cov(i, m)}{\sigma_m^2} = \rho_{im} \frac{\sigma_i}{\sigma_m}, \quad (1-2)$$

其中， $Cov(i, m)$ 为证券收益率与市场收益率的协方差， ρ_{im} 为证券收益率与市场收益率的关联系数， σ_i 、 σ_m 分别为证券收益率与市场收益率的标准差。那么，资产 i 的期望收益率 $E_i = \alpha_i + \beta_i E_m$ ，方差 $\sigma_i^2 = \beta_i^2 \sigma_m^2 + \sigma_{\epsilon_i}^2$ 。描述风险的方差被分解成了两部分， $\beta_i^2 \sigma_m^2$ 为与市场相关的系统性风险， $\sigma_{\epsilon_i}^2$ 为仅与资产本身相关的特殊风险。

对 N 个资产进行组合，并引入市场无风险利率 r_f 后，则有，

$$E(r_p) = \alpha_p + \beta_p (E_m - r_f) + r_f, \quad (1-3)$$

$$\sigma_p^2 = \sum_i^N \sum_j^N w_i w_j \beta_i \beta_j \sigma_m^2 + \sum_i^N w_i^2 \sigma_{\epsilon_i}^2. \quad (1-4)$$

其中， $\alpha_p = \sum_i^N w_i \alpha_i$ ，为组合收益中与市场无关的部分，对于包含全市场的组合 $\sum \alpha \equiv 0$ ； $\beta_p = \sum_i^N w_i \beta_i$ ，为组合的因子载荷。 α 与 β 是评价投资组合构建的重要指标 (见 1.2.3)。若组合中的资产权重相同，当资产数 $N \rightarrow \infty$ 时， $\sum_i w_i^2 \sigma_{\epsilon_i}^2 \rightarrow 0$ ，资产的特殊风险被分散，组合风险只取决于市场风险及组合的因子载荷， $\sigma_p = |\beta_p| \sigma_m$ 。

不同于 Markowitz 模型，CAPM 考虑了单个资产风险与市场风险之间的关系，将资产风险分解为系统性风险和特殊风险两部分。CAPM 假设：(1) 投资者都是理性的且具有一致预期；(2) 市场上所有资产可自由交易；(3) 市场信息完全公开；(4) 无税收无交易成本。

1.2.1.3 套利定价理论

CAPM 模型中, 组合的系统性风险由单一市场因子 $(E_m - r_f)$ 决定。Ross [15] 提出的套利定价理论 (Arbitrage Pricing Theory, APT) 进一步扩展了 CAPM, 将资产定价由单因子模型推广至了多因子模型。APT 假设组合中的任一资产 i 的收益率 $r_i = \alpha_i + \sum_j^N \beta_{ji} f_j + \epsilon_i$, 其中 f_j 为第 j 个因子的收益率, 或称风险溢价。那么组合的期望收益率,

$$E(r_p) = \alpha_p + r_f + \sum_i^N \beta_{pi} E(f_i), \quad (1-5)$$

其中, $E(f_i)$ 为组合中第 i 个因子 f_i 的期望收益率, β_{pi} 为对应的因子载荷, 取决于因子收益率与组合收益率的标准差及关联系数, $\beta_{pi} = Cov(p, i) / \sigma_i^2 = \sigma_i \sigma_p \rho_{pi}$ 。

与 CAPM 相同, 套利定价理论假设非系统性风险可以在投资组合中实现分散。当忽略了非系统性风险后, 具有相同因子载荷的资产或组合的期望收益率应该也相同, 否则市场将存在套利机会。因此, 套利定价理论是基于无套利市场得出的结论。另外, 套利定价理论虽然提出了多因子模型的理论框架, 但并未给出具体的风险因子。

1.2.1.4 FAMA 三因子、五因子模型

1992 年, FAMA et al. [16] 提出了著名的三因子模型, 在 CAPM 的基础上加入了市值因子 (Small [market capitalization] minus Big, SMB) 和账面市值比因子 (High [book-to-market ratio] minus Low, HML)。公式 1-3 变为,

$$E_p = \alpha_p + r_f + \beta_p (E_m - r_f) + s_p \cdot SMB + h_p \cdot HML, \quad (1-6)$$

其中, s_p, h_p 分别为 SMB 与 HML 的因子载荷。1997 年, Carhart [17] 扩展了 Fama 三因子模型, 加入了动量因子 ([average returns on past] Winners minus Losers, WML)。2015 年, Fama et al. [18] 又提出了五因子模型, 在三因子模型的基础上加入了盈利因子 (Robust [profitability] minus Weak, RMW) 和投资因子 ([invest] Conservatively minus Aggressively, CMA)。

1.2.1.5 多因子模型

一般来说, 因子模型可以分为三种类型^[19]:

- 宏观因子模型 (macroeconomic factor model)，采用宏观经济数据作为因子。例如：市场风险、价格指数 (CPI, PPI, 通货膨胀率)、生产总值 (GDP)、货币增长、央行利率、失业率等^[20]。原始的 CAPM 属于宏观单因子模型。
- 基本面因子模型 (fundamental factor model)，采用与公司财务有关的基本面指标作为因子。如所处行业、公司市值、市盈率、市净率等。FAMA 的三因子、五因子模型就是一种基本面因子模型。常用的还有 Barra 行业因子模型。
- 统计学因子模型 (statistical factor model)。统计因子模型利用统计学工具，从数据中挖掘和构造与资产价格相关的因子。与宏观因子模型和基本面因子模型相比，基于统计学建立的多因子模型牺牲了模型的可解释性，但带来了更多的灵活性。APT 属于统计学因子模型。

1.2.2 投资组合构建

利用多因子模型构建投资组合的过程如图1-2，一般可以分为四个阶段：

- 数据预处理。主要包含数据的清理/插补：去除或纠正数据中的错误和缺失值；数据聚合：将不同来源或不同时间点的数据进行聚合统一分析；以及数据的标准化：对数据进行规范化处理，使之处于同一量级，便于比较不同数据集；
- 因子挖掘。主要包含数据分析：对数据进行统计分析，识别可能影响投资回报的关键变量；因子设计：基于数据分析结果设计预测未来收益的因子；以及因子的选择与评估：评估各个因子的有效性，并选择最终用于模型的因子。
- 建模。主要包含模型的构建：利用选定的因子构建预测模型，这个模型旨在预测资产的期望回报；回测分析：通过历史数据测试模型的性能，验证其在过去的表现；以及压力测试：检验模型在极端市场情况下的表现和稳健性。
- 组合优化。主要包含均值-方差优化：通过优化模型来平衡预期收益和风险，寻求最优的资产配比；风险中性化：对冲不必要的风险，确保投资组合的系统性风险与市场或预定基准一致；换手控制：管理组合的买卖频率，控制交易成本。

传统的多因子模型是当前量化投资组合构建的重要形式，但并非所有的量化投资组合管理都仅基于多因子模型。特别是随着机器学习等技术的发展，导致投资组合的构建不再拘泥于线性模型。许多实证研究已经表明，采用机器学习和深度学习模型构建的投资组合，在某些情况下可以胜过多因子模型^[22-25]。

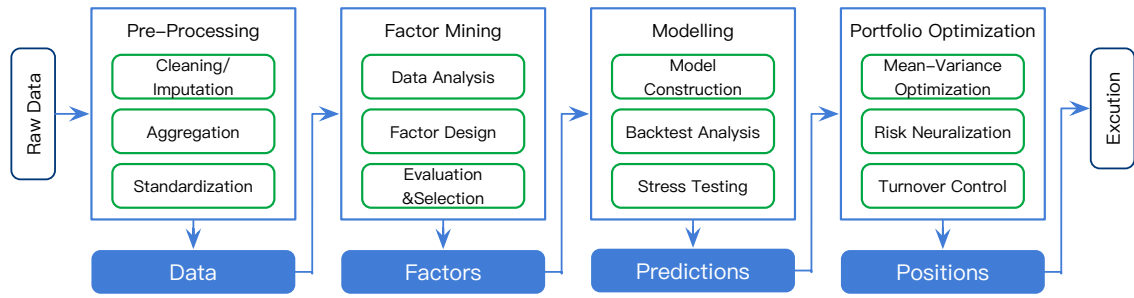


图 1-2 投资组合的构建过程^[21]。

1.2.3 评价体系

表1-3列出了评价投资策略的常用指标，及其含义和计算公式。在进行策略评估时，需要综合考虑各个指标的影响。

指标名称	含义	计算公式
Alpha	组合的超额收益	$\alpha_p = r_p - r_m$
Beta	组合收益相对市场波动	$\beta_p = \frac{Cov(p, m)}{\sigma_m^2} = \rho_{pm} \frac{\sigma_p}{\sigma_m}$
Sharpe ratio ^[26-27]	单位总风险的收益率	$SR_p = \frac{r_p - r_f}{\sigma_p}$
M2 Measure ^[28]	考虑杠杆时的收益率	$M2_p = r_f + \sigma_m SR_p$
Treynor ratio ^[29]	单位系统风险的超额收益率	$TR_p = \frac{r_p - r_f}{\beta_p}$
Information ratio ^[30]	单位主动风险的超额收益率	$IR_p = \frac{r_a - r_f}{\sigma(r_p - r_f)}$
Jensen's Alpha ^[31]	实际收益率与理论收益率之差	$\alpha_p = r_p - [r_f + \beta_p(r_m - r_f)]$

表 1-3 投资策略的评价指标。

1.3 算法交易

量化投资的另一个核心部分是算法交易。早期的股票和其他金融工具的交易通常是在交易所的交易厅内进行，交易员需要通过电话、传真或者面对面的方式进行现场交易。然而，这些传统交易方式不仅效率低下，还容易受到信息不透明和

人为操纵的影响。

1971 年，美国纽约诞生了首个电子交易所——纳斯达克（Nasdaq）。电子交易平台和电子通讯网络（ECN）的出现，使得交易员可以通过计算机进行交易，无需亲临交易所。到了 1990 年代，几乎所有的证券交易都转移到了更快速、更高效的电子交易平台，这为算法交易的兴起创造了必要的条件。

市场参与者为了降低交易成本、抵御市场冲击，开始采用计算机制定的交易策略，并通过互联网完成交易，这极大地提升了市场交易效率。做市商市场交易规则的改变和北美开放市场中的套利机会，进一步推动了高频交易的发展^[32-33]。高频交易的兴起，又进一步改变了市场结构^[34]。如今，大多数量化投资策略都是通过计算机算法在市场上执行交易的。

与传统的人工交易方式相比，算法交易具有多重优势，例如：

- 高效的执行效率：算法交易能够在纳秒级别内执行交易，降低人工交易成本，并且避免人工交易犯错的可能性。
- 快速的决策速度：算法交易可以迅速识别和利用市场变动，从而避免错过有利的交易时机。
- 较低的情绪影响：算法交易能够严格执行交易策略，从而避免人为因素给策略带来的情绪性干扰。
- 多策略的综合运用：算法交易可以同时应用多种交易策略，这些策略可以基于市场数据、统计模型、技术分析等多种因素，从而降低单一策略带来的风险。
- 实时风险管理：算法交易可以实时监控市场变动，从而进行及时的风险管理。

虽然算法交易具备以上优势，但是自其诞生以来也一直伴随许多争议^[35]，争议的主要内容包括：对市场流动性的影响^[4,36-37]，在市场波动时高频交易会不会引起流动性损失；对价格波动的影响^[38-39]；是否使价格趋近其真实价格，从而使市场更有效；高频交易者的利润是否来源于传统交易者，从而损害了传统交易者的利益，高频交易是否是知情交易^[40]；算法交易是否会引起或加剧市场闪崩，从而在市场危机时增加流动性损失的风险^[41]；市场是否会由于算法交易的引入而更容易出现价格操纵；市场的组织形式是否应该针对高频交易者进行更改、对数据的披露是否应该更加公开、对高频交易者是否应该单独制定收费机制等。

综上，算法交易的发展，不仅展现了技术进步给金融市场带来的机遇，也突显了其在市场稳定性和透明度方面的新挑战。

1.3.1 市场微观结构

市场微观结构是一门研究交易在金融市场内的组织形式和详细机制的学科^[42-44]。通过对所有市场参与者在市场内发生的粒度级别的交互和动态进行分析，市场微观结构试图解释价格形成和发现、分析交易成本、并完善市场的组织结构等。了解市场微观结构对于算法交易策略的制定至关重要，

1.3.1.1 交易数据的组织形式——限价订单簿

市场微观结构的研究建立在粒度级别的交易数据之上，其最重要的组织形式是限价订单簿（Limit Order Book, LOB）。

交易所接受的订单通常可以分为两类：市价订单（Market Order）与限价订单（Limit Order）。市价订单是指以当前市场上的最优价格立即成交的订单，限价订单是指以不劣于交易员规定的价格成交的订单。限价订单簿就是一种系统化排列限价订单的数据组织形式。

图1-3是限价订单簿在某一时刻的快照示意图。绿色、红色分别为买方、卖方委托量随报价的分布。买方最高报价与卖方最低报价之差为买卖差价（ask-bid spread）。当新的限价委托提交时，会排列到相应价格的队列之后。成交时一般采用“先进先出”的模式（FIFO, First In First Out），即同样的价格先委托的先成交。当市价单到达时，会根据限价单的最优报价进行撮合，当最优报价全部成交后，市价将产生变动。



图 1-3 限价订单簿。

A 股交易数据的组织形式 中国有上海和深圳两个证券交易所，其数据组织形式大体相同。上交所提供 Level-1 和 Level-2 两档数据，对应深交所的基本行情与增

强行情。Level-2 的实时行情包含快照类数据以及逐笔数据。其中，逐笔成交数据包括每一笔成交的证券代码、成交时间、成交价格、成交数量、成交金额、成交序号、成交通道、买方订单号、卖方订单号以及内外盘标志。逐笔委托数据包括每一笔委托的证券代码、委托时间、委托价格、委托数量、委托序号、委托通道、原始订单号以及内外盘标志。快照类数据（限价订单簿）每 3 秒更新一次，包括了每支证券的代码、昨收盘价格、开盘价格、最高价格、最低价格、最新价格、收盘价格、成交笔数、成交量、成交金额、委托买入总量、加权平均委买价格、委托卖出总量、加权平均委卖价格、买入总笔数、卖出总笔数、买卖 10 档价格和数量等¹。

ID	Time	Price	Ask Price	Ask Volume	Bid Price	Bid Volume	Volume	Turnover
600703.SH	09:35:11	¥324.70	¥324.80	700	¥324.60	500	7123769	¥232,356.81
600703.SH	09:35:14	¥325.00	¥325.00	200	¥324.90	4500	7176769	¥234,078.39
600703.SH	09:35:17	¥325.00	¥325.30	400	¥325.00	1200	7217769	¥235,410.71
600703.SH	09:35:20	¥325.40	¥325.50	17000	¥325.40	12400	7242469	¥236,213.71
600703.SH	09:35:23	¥325.60	¥326.00	200	¥325.60	11800	7290369	¥237,773.62
600703.SH	09:35:26	¥326.20	¥326.30	600	¥326.10	4800	7373469	¥240,479.98

表 1-4 限价订单簿样本数据。

A 股市场的特点 A 股市场与全球其他股票市场的重要区别之一是采取次日交易日起回转交易，即“T+1 交易规则”。该规则不允许投资者在同一天卖出当天买入的股票。但是在持有底仓的情况下，仍然可以在效果上实现“T+0”交易。A 股市场的另一个显著特点是设置了 10% 到 20%（科创板）的涨跌停阈值，某支股票达到该阈值后会强制停止竞价。此外，A 股市场不可以直接卖空股票，无法直接利用做空进行风险对冲（需通过融资融券）。上述规则在一定程度上抑制了市场上的投机行为，保护了 A 股以“散户”为主的大部分参与者的利益^[45]。

1.3.1.2 市场微观结构理论

随着金融市场的不断开放与完善、市场交易数据的不断累积、以及数据形式的不断增加，市场微观结构理论的复杂性在不断提高，因此，目前该领域并没有一个被广泛认可的模型。

市场微观结构的研究可以分为三代^[46]。第一代模型是价格序列模型，仅采用了价格信息，较早期的模型有交易分类模型和 Roll 模型。第二代模型增加了交易量

¹ 深圳证券交易所：<http://www.szse.cn/marketServices/index.html>

上交所信息网络有限公司：<https://www.sseinfo.com/services/assortment/document/>

数据，研究的方向转移到了交易量对价格的影响，Kyle [47]和 Amihud [48]是这一代的两个代表性模型。第三代模型始于知情交易概率理论（Probability of Informed Trade, PIN）的建立^[49-50]，PIN 理论认为买卖差价是做市商（提供流动性）和知情交易者（持有头寸）之间连续性博弈的均衡结果。

假设证券价格为 S ，现价为 S_0 。价格受到新信息的影响后变为 S_B （坏消息）与 S_G （好消息）。假设新信息的到达率为 a ，是坏消息的概率为 δ ，是好消息的概率为 $1 - \delta$ 。那么 t 时刻，价格的期望值为，

$$E(S_t) = (1 - \alpha_t)S_0 + \alpha_t[\delta_t S_B + (1 - \delta_t)S_G] . \quad (1-7)$$

若报单行为服从泊松分布，知情交易者的报单到达率为 μ ，非知情交易者的报单到达率为 ϵ ，那么，做市商与知情交易者之间经过博弈后，达到买卖差价的盈亏平衡点为，

$$E(A_t - B_t) = \frac{\mu\alpha_t(1 - \delta_t)}{\epsilon + \mu\alpha_t(1 - \delta_t)}(S_G - E(S_t)) + \frac{\mu\alpha_t\delta_t}{\epsilon + \mu\alpha_t\delta_t}(E(S_t) - S_B) . \quad (1-8)$$

当 $\delta_t = 1/2$ 时，买卖差价的盈亏平衡点为，

$$E(A_t - B_t) = \frac{\alpha_t\mu}{\alpha_t\mu + 2\epsilon}(S_G - S_B) . \quad (1-9)$$

式中， $(S_G - S_B)$ 的系数决定是盈亏平衡点的关键因素，定义为知情交易概率 PIN，

$$PIN_t = \frac{\alpha_t\mu}{\alpha_t\mu + 2\epsilon} \quad (1-10)$$

通过拟合 $(\alpha, \delta, \mu, \epsilon)$ 可以得到 PIN 的值。

1.3.1.3 算法交易的认定与监管

对于算法交易，国内市场的监管还处于探索阶段。在这方面，德国最早对高频交易实施了较为严格的监管。德国监管机构要求高频交易必须备案，禁止利用高频交易进行试探性交易（Ping Orders）、误导簇交易（Quote Stuffing）、引发动量交易（Momentum Ignition）、分层与欺诈交易（Layering and Spoofing）等。国内期货市场对高频交易的判定采取 3+3 或 5+5 规则，即投资者在 1 秒内有 3（或 5）次报单，且当天重复出现 3（或 5）次以上，则被认定为高频交易。但是，在股票市场尚无明确认定方式。

1.3.2 交易成本分析

交易成本分析 (Transaction Cost Analysis, TCA) 是评估交易策略及执行流程效率和有效性的重要工具。

1.3.2.1 交易成本构成

交易成本包含直接成本和间接成本两部分^[51-57]。

直接成本是与执行交易直接相关的明确、可测量的费用，包括佣金 (commissions)，交易所费用，税金等。这些费用是在交易发生后，由交易所、交易商及税务部门直接收取的费用。A 股市场的佣金对个人交易者通常在 2-3bps，单笔最低 5 元，对机构交易者可以更低。印花税为 10bps，2023 年底减半征收，目前为 5bps，仅对卖家征收。此外，上海证券交易所还会对每笔交易征收 0.1bps 的过户费。

间接成本是不太显而易见的成本，也是交易成本分析的主要研究对象。间接成本主要包含：(1) 市场冲击 (Market Impact)。市场冲击是指交易自身导致的价格向交易方向不利的变动。(2) 机会成本 (Opportunity Cost)。机会成本是由于流动性丧失导致无法交易时产生的成本。(3) 时间延迟成本 (Delay Cost)。时间延迟成本是由于时间延迟导致成交前市场价格发生变动时产生的成本。(4) 价格漂移成本 (Price drift)。价格漂移是指证券价格在交易期内发生的异动，即该时间段内的 α 。价格漂移成本是在证券价格异动时买卖股票造成的额外成本。(5) 择时风险 (Timing Risk)。择时风险包含了价格波动和流动性风险，波动异常或流动性降低都会显著影响交易成本。(6) 买卖差价 (ask-bid spread)。当交易者需要在市场上立即成交时，通常需要以支付买卖差价作为代价。这些间接成本虽然不会立即显现，但可以显著影响交易绩效。

对于大宗交易，市场冲击是影响交易成本最重要的因素。市场冲击的种类并非单一，其构成较为复杂。通常可以将市场冲击分为临时市场冲击和永久市场冲击两类^[58]。临时市场冲击是交易者在市场上获取流动性所产生的冲击，流动性越低，临时冲击越大；永久市场冲击是由于交易者的交易意图暴露在市场导致的永久性价格变动，通常表现在订单簿的买卖失衡，买卖失衡直接影响供需关系，从而永久改变证券价格。除了构成复杂，市场冲击的形式在不同市场间也存在差异^[59-65]。上面的各种因素造成了市场冲击理论的多样与复杂^[66-67]，不仅如此，有实证研究表明市场冲击理论与实际交易之间存在一定差距^[68-69]。

1.3.2.2 交易执行缺口

交易成本分析中的一个重要概念是执行缺口^[70] (Implementation Shortfall, IS)。执行缺口是预期回报 (Paper Return) 与实际回报 (Actual Return) 之差。假设待交易的股票数量为 S ，预期回报由决策价格 (decision price) P_d 和交易结束时的市场价格 P_n 决定，

$$PR = S(P_n - P_d) \quad (1-11)$$

实际回报，

$$AR = \sum s_i P_n - \sum s_i p_i - fees, \quad (1-12)$$

其中， s_i ， p_i 分别为第 i 笔交易的数量及价格， $fees$ 代表交易产生的固定费用。则执行缺口为，

$$IS = S(P_n - P_d) - \sum s_i (P_n - p_i) + fees. \quad (1-13)$$

(1) 如果交易全部完成，则有， $\sum s_j = S$ 。此时的执行缺口，

$$IS = S(P_a - P_d) + fees, \quad (1-14)$$

其中， $P_a = \sum s_i p_i / S$ 为交易执行的平均价格。

(2) 如果交易没有全部完成， $\sum s_i \neq S$ ，此时的执行缺口可以写为，

$$IS = \sum s_i (P_a - P_d) + \left(S - \sum s_i \right) (P_n - P_d) + fees \quad (1-15)$$

式中第一项为交易的执行成本，第二项是由于交易未完成所产生的机会成本。

(3) 考虑到决策与交易间的延迟，决策价格与到达价格 (Arrival Price) 之间通常存在差异。改写 $P_n - P_d = (P_n - P_0) + (P_0 - P_d)$ ，其中 P_0 为到达价格。此时的执行缺口可以写为，

$$IS = S(P_0 - P_d) + \sum s_i (P_a - P_0) + \left(S - \sum s_i \right) (P_n - P_0) + fees \quad (1-16)$$

公式1-15中的交易执行成本和机会成本中，各有一部分被分解合并到了延迟造成的成本中，即上式第一项。第二项、第三项对应修正后的交易执行成本和机会成

本。

(4) 通常决策价格是由组合经理决定的，与算法交易和交易执行期间的市场活动本身无关，剔除这部分影响后的执行缺口为，

$$IS = \sum s_i(P_a - P_0) + \left(S - \sum s_i\right)(P_n - P_0) + fees \quad (1-17)$$

式中第一项为交易执行成本，第二项为机会成本。

1.3.2.3 交易执行结果评估

公式1-17中的第一项是与交易执行直接相关的成本，代表交易者希望以开始下单时的市场价格(到达价格)成交时，所需支付的交易执行成本。在限价订单簿中，到达价格通常取当时买卖价格的中间值。这里的到达价格可以看作交易选取的基准价格。更一般的，交易成本(trading cost, TC)取决于交易执行价格(execution price)与基准价格(benchmark price)之差。以基点(base points, bps)为单位的交易成本为，

$$\text{trading cost} = \text{side} \cdot \frac{\text{execution price} - \text{benchmark price}}{\text{benchmark price}} \times 10,000 \text{ (bps)} . \quad (1-18)$$

其中，交易方向为买时，side = +1，交易方向为卖时，side = -1。对于订单拆分，交易执行价格为平均价格 P_a 。基准价格可以根据不同的交易和分析目标进行选取，例如到达价格(Arrival Price)、TWAP、VWAP、收盘价格等。

目前市场上最常用的基准价格是 VWAP (Volume-weighted average price)。VWAP 是按照证券交易量加权后的市场平均价格：

$$VWAP = \sum_i P_i V_i / \sum_i V_i . \quad (1-19)$$

但是，使用 VWAP 存在一些局限^[71]：首先，订单越大，结果会越接近 VWAP 价格。极端情况，当交易时段市场参与度为 100% 时，那么交易的结果就等于 VWAP 价格。其次，当在不同交易所进行大宗交易，特别是交易者仅有有限机会参与交易时，用 VWAP 作为基准，结果可能会有偏差。第三，VWAP 的结果会受到波动率等因素的较大影响，因此不便于比较不同股票、或同一股票在不同日期间的交易结果。

参与度加权价格基准 (Participation-Weighted Price, PWP) 是按照市场参与度，

计算得到的对应市场成交量的 VWAP:

$$PWP = \text{market VWAP for volume } V, \text{ where } V=S/POV. \quad (1-20)$$

其中, S 代表需要完成的交易量, POV 为预设的市场参与度, V 是交易开始时到市场累计完成 $V=S/POV$ 时的交易量。与 VWAP 基准类似, PWP 同样不适合进行跨股或跨日的比较。此外, 投资者有可能通过更激进的交易操纵 PWP。由于市场冲击的影响, 投资者可以通过在短时间内推高买单(推低卖单)价格, 造成后续仍有大量交易的假象, 并将价格维持在较高(较低)水平。此时, PWP 需要更长的等待时间以满足市场交易量的需求, 导致计算出的 PWP 向对投资者更有利的方向移动, 而无法反应订单在实际交易时的市场价格。

上面的基准都包含了市场或行业的总体波动, 在市场上升或下降时交易股票会使计算的交易成本偏离预期。经过市场调整后的成本, 可以将交易执行成本与证券价格随总体市场波动的部份分开^[72]。

$$\text{adjusted trading cost} = \text{trading cost} - \beta \times \text{market cost}. \quad (1-21)$$

其中, β 为证券对市场的载荷, 市场成本为,

$$\text{market cost} = \text{side} \cdot \frac{\text{market VWAP} - \text{market arrival price}}{\text{market arrival price}} \times 10,000 \text{ (bps)}. \quad (1-22)$$

市场成本通常选择相应的市场或行业指数进行计算。

1.3.3 传统算法交易策略

假设算法需要在交易时长 T 内完成交易量 V , 为了完成交易并降低市场冲击, 交易者需要将订单进行拆分。传统的拆分方式有时间加权平均价格(Time-weighted Average Price, TWAP)、交易量加权平均价格(Volume-weighted Average Price, VWAP), 成交量百分比(Percent Of Volume, POV)等。

1.3.3.1 TWAP

TWAP 是交易量随时间均匀分布时, 所有成交价格的简单算数平均。假设订单在交易时间段内被均匀拆分为 N 份, 且拆分后第 i 份子订单的成交价格为 P_i , 那

么该订单的时间加权平均成交价格为,

$$TWAP = \frac{1}{N} \sum_{i=1}^N P_i . \quad (1-23)$$

虽然将订单等时拆分很方便,但这种交易方式比较容易被交易对手嗅探。因此,为了不被市场察觉,交易者往往需要将订单拆分到足够小。此外,也有算法对交易时间甚以及交易量在 **TWAP** 的基础上加上随机扰动,以隐藏交易意图。

1.3.3.2 VWAP

全市场的 **VWAP** 常被看作公平市场价格 (fair market price), 因此常被用于算法交易的基准价格。**VWAP** 算法的实质是追踪全市场的 **VWAP**。假设订单被拆分为 N 份, 每份成交量为 V_i , 成交价格为 P_i , 那么该订单的交易量加权平均成交价格为,

$$VWAP = \sum_{i=1}^N P_i V_i / \sum_{i=1}^N V_i . \quad (1-24)$$

其中每份成交量 V_i 通常根据对应时间段内的历史成交量曲线, 按照一定比例进行分配。

由于真实市场的交易量在随时变化, 实现 **VWAP** 难点之一是市场交易量分布的计算和预测。另外, 交易频率的选择也至关重要, 合理的交易频率的可以有效地隐藏自己的交易意图, 降低市场冲击。**VWAP** 算法有许多变体 (例如: Konishi [73], Gomber et al. [74], Biakowski et al. [75], McCulloch et al. [76], Frei et al. [77], Mitchell et al. [78], Busseti et al. [79], Cartea et al. [80], Barzykin et al. [81]), 是目前市场上应用最广泛的算法之一。

1.3.3.3 POV

市场参与度 **POV** (Percent of Volume) 是指按照市场当前成交量的固定百分比参与交易的算法。与 **TWAP** 一样, **POV** 一般将交易时段进行等时划分, 其成交量由市场动态调节。对于第 i 期交易, **POV** 算法参与的订单量 S_i 为,

$$S_i = POV \cdot (V_{market,i-1} - S_{i-1}) . \quad (1-25)$$

其中，POV 为预设的百分比， $V_{market,t-1}$ 为上一期市场的交易量， S_{t-1} 为上一期算法自身成交量。POV 算法在执行大量订单时可以有效隐藏自己的交易信息，也是目前被广泛使用的算法之一。

上述介绍的三种算法都是被动算法。图1-4展示了这三种算法相对市场交易量随时间的分布示意图。他们的实现相对容易，但可能损失的机会成本较大。

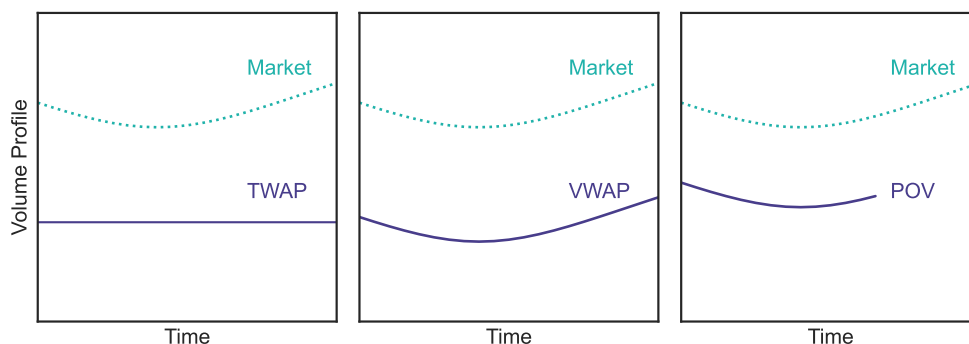


图 1-4 TWAP、VWAP 及 POV 相对市场交易量随时间的分布示意图。

1.3.3.4 其他算法

除了上述算法，市面上还有针对私人交易所设计的暗池算法^[82]（“Dark Pool”），隐藏部份订单量的冰山算法^[83]（“Iceberg”），配对交易算法^[84]（Pair Trading）等。

除了这些传统算法，随着人工智能的发展，越来越多的交易者开始借助机器学习、深度学习和强化学习等技术制定自己的交易策略。

第二章 机器学习与算法交易

人工智能（Artificial Intelligence, AI）是一个研究如何使计算机系统具备智能行为和思维能力的领域。作为学术概念，人工智能可以追述到 1950 年代^[85]。早期对人工智能的研究集中在符号逻辑、推理和问题求解上。到 1980 年代，机器学习开始引领人工智能的发展，支持向量机^[86-87]（Support Vector Machine, SVM）、决策树^[88]（Decision Tree, DT）和神经网络^[89-90]（Neural Network, NN）等算法的发展推动了机器学习领域的进步，深度学习的概念也初现雏形。2010 年后，得益于显卡运算能力的快速提升，基于大数据的深度学习模型取得了突破性的成果，由此人工智能开始在各个领域被广泛应用。

在量化投资领域，人工智能已经被应用到各个方向。例如，将线性资产定价模型推广到非线性的资产定价模型^[91-94]，构建并优化投资组合^[95]，预测资产价格^[96-98]，管理投资风险^[95]，分析市场情绪^[99-101]，进行市场合规检测^[102] 等等。本章将简要介绍若干常用的机器学习模型，以及目前人工智能在量化投资领域中的几个重要应用。

2.1 机器学习简介

传统的计算机算法在处理问题时需要基于预设的规则（rule-based），当问题变得越来越复杂时，编写这类算法会遇到前所未有的阻力，甚至无法实现。机器学习则提供了一套基于经验（experience-based）解决思路，通过对大量数据的学习令算法建立自身的规则，从而依据经验对未知数据做出推断。

图2-1展示了机器学习的类型与常见算法。机器学习根据学习方式的不同大致可以分为有监督学习、无监督学习以及强化学习。根据解决的问题类型又可分为回归问题以及分类问题（分类问题也可以抽象为目标为若干离散值的回归问题）。深度学习是基于神经网络（Walter Pitts and Warren McCulloch, 1943）发展起来的一大类机器学习类型。另外，集成学习近年也是一大类被广泛应用的机器学习思想。

2.1.1 传统机器学习

传统机器学习是基于统计与代数发展起来的一类算法，需要人工提取数据的特征（特征工程、特征选取、数据清洗），然后进行建模。

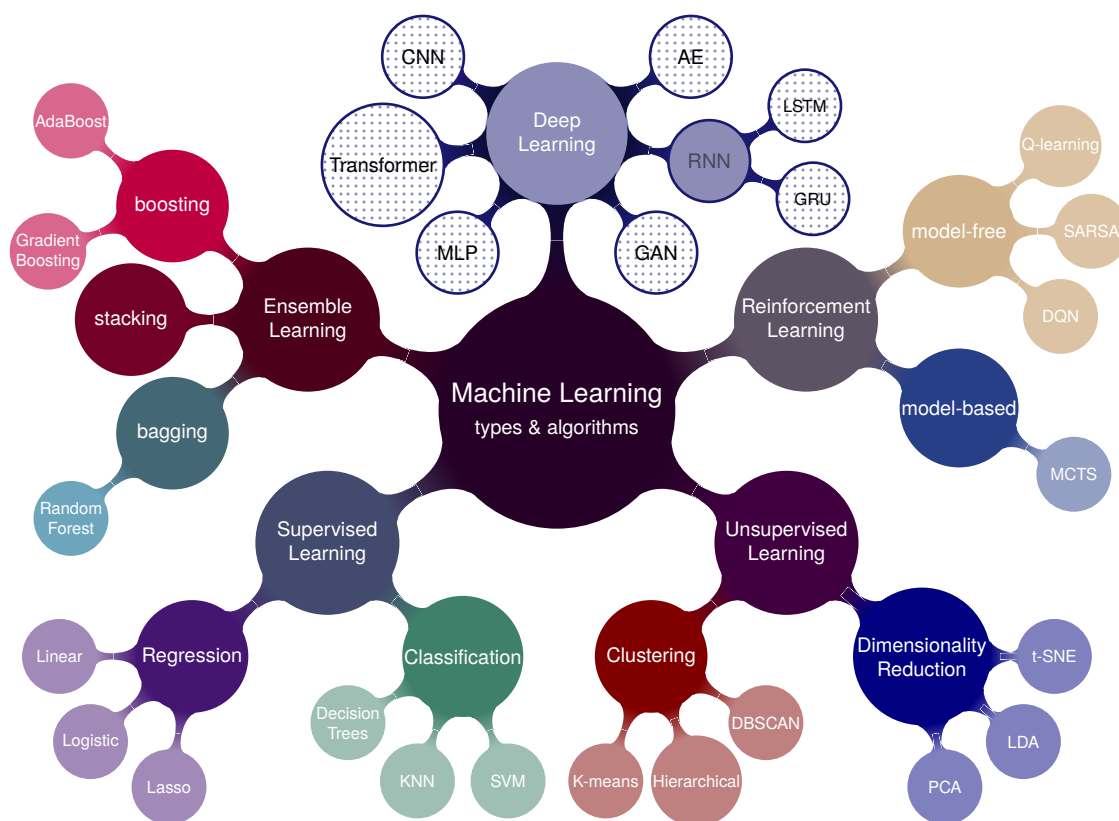


图 2-1 机器学习的类型与代表性算法。

2.1.1.1 线性回归

线性回归模型是最简单的机器学习模型，也是许多复杂模型特别是深度学习的基础。线性回归的原理是利用线性函数对数据进行拟合，

$$f(x) = wx + b. \quad (2-1)$$

其中，自变量为 x ， w 称为权重， b 称为偏置。假设有 N 个训练样本 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ ，定义损失函数（loss function）为样本真值与预测值间的误差，成本函数（cost function）为样本的平均损失函数。对线性回归通常取均方误差为模型的成本函数，

$$J(w, b) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2. \quad (2-2)$$

训练的目标是找出最优参数 $\hat{w}$ 与 $\hat{b}$ 使成本函数最小。

梯度下降法（Gradient descent）是求解优化问题的常用方法之一。

$$\hat{w} = w - \eta \frac{\partial J}{\partial w}, \hat{b} = b - \eta \frac{\partial J}{\partial b} \quad (2-3)$$

其中， η 为人为设定的机器学习的学习速率或步长（learning rate/step size）。对上式进行迭代直至结果收敛，得到最优参数 $\hat{w}$ 与 $\hat{b}$ 。

当样本数与特征值数目相等时，称为充分参数化（sufficiently parameterized），此时仅有一个最优解，对应于成本函数等于 0 的解。当样本数大于特征数目时，称为欠参数化（under-parameterized），此时找不到成本函数等于 0 的解，最优解对应于成本函数梯度等于 0 的解。当样本数小于特征数目时，称为过参数化（over-parameterized），此时可以找到多个最优解，由梯度下降法给出的解称为 Moore-Penrose 解。

线性模型同样可以运用于分类问题。例如，线性判别分析（Linear Discriminant Analysis, LDA）是用于二分类问题的线性模型^[103]。

2.1.1.2 支持向量机

支持向量机是一类监督学习方法，由统计学习演化而来^[86-87,104]。依然考虑一个二分类问题，假设有一组训练数据 $\{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ ，其中 $x \in \mathbb{R}$, $y \in (1, -1)$ ，1 代表正样本，-1 代表负样本。线性分类器的学习目标是找到一个超平面 $h(x) = wx + b = 0$ ，使其可以区分所有正负样本。支持向量定义了一个距离正负样本几何间隔（margin）最大的超平面。

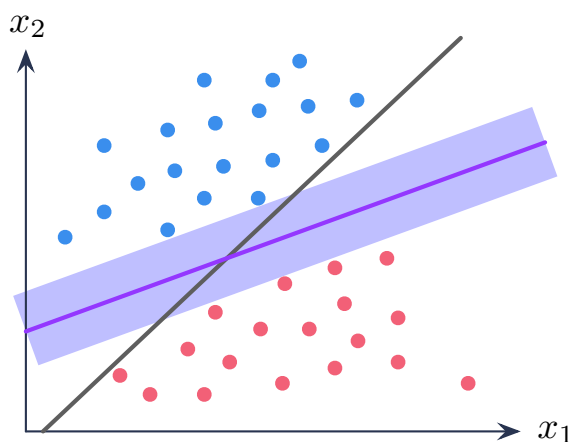


图 2-2 支持向量机示意图。

求解支持向量可以转化为如下的凸优化问题：

$$\min \quad \frac{1}{2} \|w\|^2 \quad (2-4)$$

$$\text{s.t.} \quad y_i(w x_i + b) \geq 1, i = 1, \dots, \ell \quad (2-5)$$

其中, $\|w\| = \langle ww \rangle$, $\langle ab \rangle$ 代表 a 点乘 b 。对于有约束的凸优化问题, 可以使用拉格朗日乘子法。构造如下拉格朗日量,

$$L(w, b, \lambda) = \frac{1}{2} \|w\|^2 - \sum_i^\ell \lambda_i [y_i(w x_i + b) - 1] . \quad (2-6)$$

其中, λ_i 为拉格朗日乘子, $\lambda_i \geq 0$ 。分别令 $\partial_w L \equiv 0$, $\partial_b L \equiv 0$ 可得,

$$w - \sum_i^\ell \lambda_i y_i x_i = 0, \quad \sum_i^\ell \lambda_i y_i = 0 . \quad (2-7)$$

将公式2-7代回公式2-6可得,

$$L(w, b, \lambda) = -\frac{1}{2} \sum_{i,j=1}^\ell \lambda_i \lambda_j y_i y_j \langle x_i x_j \rangle + \sum_i^\ell \lambda_i \quad (2-8)$$

此时, 问题转化为了求解如下对偶优化问题,

$$\max \quad -\frac{1}{2} \sum_{i,j=1}^\ell \lambda_i \lambda_j y_i y_j \langle x_i x_j \rangle + \sum_i^\ell \lambda_i \quad (2-9)$$

$$\text{s.t.} \quad \lambda_i \geq 0, i = 1, \dots, \ell \quad (2-10)$$

$$\sum_i^\ell \lambda_i y_i = 0 . \quad (2-11)$$

根据 Karush-Kuhn-Tucker (KKT) 条件^[105-106],

$$\lambda_i [y_i h(x_i) - 1] = 0 . \quad (2-12)$$

对任意样本都有 $\lambda_i = 0$ 或者 $y_i h(x_i) = 1$ 。

根据公式2-7，求解的超平面可以写为，

$$h(x) = wx + b = \sum_{i=1}^{\ell} \lambda_i y_i \langle x_i, x \rangle + b \quad (2-13)$$

上式称为支持向量展开（SV expansion）。当 $\lambda_i = 0$ 时，样本对 $h(x)$ 没有贡献；当 $\lambda_i > 0$ 时， $y_i h(x_i) = 1$ ，此时对应的样本为支持向量。

上述讨论基于样本线性可分的前提，对于线性不可分的情形，可按照上述步骤将模型进行推广。一种常用的方式是利用核函数（kernel function）对数据进行升维，使其在更高维度上线性可分^[104,107]。

由于支持向量机模型的复杂度仅取决于支持向量的个数，因此支持向量机对小样本情况依然有效，并且通过核函数，支持向量机可以高效的学习高维数据，避免了“维度灾难”。

2.1.1.3 决策树

决策树（Decision Tree）是一类重要的监督学习算法^[88]。决策树通过对数据每个特征进行逐层划分，最终得到一个多节点的分层树形结构。

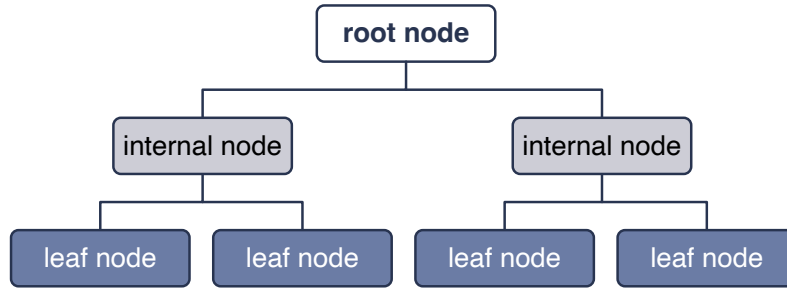


图 2-3 决策树模型示意图。

决策树模型的关键是在每个节点应该选择哪种特征对数据进行划分。选择特征的好坏可以通过不纯度（impurity）进行度量。划分后，一个分支的所有实例所属分类越一致，那么这个节点的划分就越纯。假设决策树共有 N 个实例待划分，到达节点 m 有 N_m 个实例，其中属于类别 i 的实例个数为 N_m^i ， $\sum_i N_m^i = N_m$ 。定义该节点的熵（entropy），

$$\mathcal{I}_m = - \sum_{i=1}^K p_m^i \log_2 p_m^i. \quad (2-14)$$

其中,

$$p_m^i = \frac{N_m^i}{N_m} . \quad (2-15)$$

熵值越小, 节点的纯度越高。对于二分类问题 ($K = 2$), 若 $p_m^1 = 1, p_m^2 = 0$, 此时熵最小, 为 0。对于 K 分类问题 ($K > 2$), 可知当 $p^i = 1/K$ 时熵最大, 为 $\log_2 K$ 。如果一个节点不纯, 说明该节点需要依据实例特征进一步划分。划分的依据是保证划分后子节点的熵值尽可能低。对节点 m , 假设划分后一共有 L 个子节点, N_m 中有 N_{mj} 个实例被划分到了子节点 j , 那么划分后总的纯度为,

$$I'_m = - \sum_{j=1}^L \frac{N_{mj}}{N_m} \sum_{i=1}^K p_{mj}^i \log_2 p_{mj}^i . \quad (2-16)$$

其中,

$$p_{mj}^i = \frac{N_{mj}^i}{N_{mj}} . \quad (2-17)$$

构建决策树时, 先对节点上的实例选取不同特征进行划分, 然后利用公式2-16比较这些划分后的纯度, 取纯度最低时对应的特征为该节点最终的划分依据。

熵不是纯度的唯一度量。以二分类问题为例, 令 $p^1 \equiv p, p^2 = 1 - p$, 如果一个非负函数 ψ 满足如下三点条件, 那么就可以作为一个纯度度量^[108]:

$$\begin{cases} \psi(1/2, 1/2) \geq \psi(p, 1-p), \forall p \in [0, 1], \\ \psi(0, 1) = \psi(1, 0) = 0, \\ \psi(p, 1-p) \text{ 在区间 } [0, 1/2] \text{ 递增, } [1/2, 1] \text{ 递减} . \end{cases} \quad (2-18)$$

对于熵: $\psi(p, 1-p) = -p \log_2 p - (1-p) \log_2 (1-p)$ 。常用的纯度度量还有基尼系数 (Gini index): $\psi(p, 1-p) = 2p(1-p)$; 分类误差率: $\psi(p, 1-p) = 1 - \max(p, 1-p)$ 。ID3 决策树使用熵作为纯度度量^[88], CART (Classification and Regression Trees) 使用基尼系数作为纯度度量^[109],

决策树具有简单易用、易可视化等优点, 但是随着决策树节点数和深度的不断增加, 模型容易存在过拟合的风险。一种降低过拟合风险的方式是剪枝 (pruning)。剪枝的方式有预剪枝 (pre-pruning) 与后剪枝 (post-pruning)。预剪枝是在训练决策树时, 当子节点样本数小于预设值时, 则停止划分。后剪枝是当全部训练完成后,

从叶节点开始逐层向上替代子节点，如果决策树的性能可以提升，则将子节点替换为叶节点。对比预剪枝和后剪枝，预剪枝速度更快，而后剪枝往往更精确^[110]。

2.1.2 深度学习

深度学习是机器学习领域中最热门的分支之一。传统的机器学习技术受限于需要专业领域从业者对原始数据进行特征提取。而深度学习可以直接通过对原始数据的逐层抽象表示，学习非常复杂的非线性关系^[111]。

2.1.2.1 深度学习的发展历史

Walter Pitts 和 Warren McCulloch 在 1943 年提出了神经网络最初的数学模型^[112]。Rosenblatt 在 1957 建立了可以实际应用于二分类问题的感知机（Perceptron）模型^[113]。1960 年，Henry J. Kelley 提出了第一个反向传播模型^[114]。1969 年，Marvin Minsky 和 Seymour Papert 出版了“Perceptrons”一书^[115]，指出 Rosenblatt 的感知机模型存在若干缺陷，例如无法解决异或运算（XOR）。这导致人工神经网络在 1970 年代的发展非常缓慢。

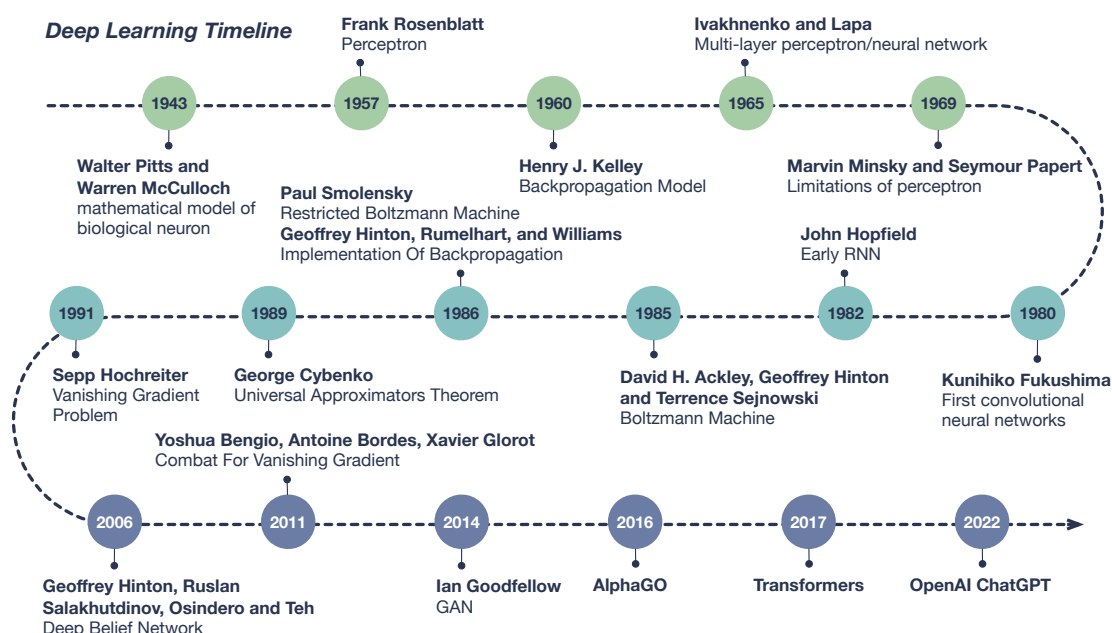


图 2-4 深度学习发展时间线。

直到 1986 年，Geoffrey Hinton, Rumelhart, 和 Williams 成功将反向传播模型运用于神经网络中，使得训练复杂的神经网络变得可行^[116]。1989 年，Yann LeCun 及

合作者利用反向传播（Back-Propagation, BP）算法训练了卷积神经网络，并成功运用于手写字体的识别^[117]，这为后续深度学习在计算机视觉领域的发展奠定了基础。同年，George Cybenko 提出了通用近似定理（Universal Approximation Theorem）的雏形，证明拥有隐层及有限神经元个数的前馈神经网络有能力逼近任何连续性方程^[118]。但是受限于当时计算机的处理速度，训练深度神经网络还无法实现。此外，增加神经网络的规模可能导致的“梯度消失”问题进一步限制了神经网络的应用。深度学习的发展第二次受阻。

2006 年，Geoffrey Hinton, Ruslan Salakhutdinov, Osindero 以及 Teh 在其称为深度信念网络（Deep Belief Networks）中，堆叠了多个受限玻尔兹曼机（Restricted Boltzmann Machine, RBM），使得模型对大数据的训练效率大幅提高^[119]。之后，随着计算机硬件性能的提升，特别是人们发现 GPU 在处理神经网络算法上的巨大优势，真正开启了深度学习的时代。

2011 年，Yoshua Bengio, Antoine Bordes 和 Xavier Glorot 的工作展示了使用 ReLU 作为激活函数可以有效避免“梯度消失”的问题^[120]。2014 年，Ian Goodfellow 及合作者构造了生成对抗网络（Generative Adversarial Neural Network, GAN）^[121]。GAN 以假乱真的能力，使其迅速被运用在了科学、艺术、时尚等各个领域。2016 到 2017 年，AlphaGo 战胜一众围棋世界冠军，展示了其强大的实力。2017 年，谷歌团队提出了 Transformer 模型^[122]，并在 2018 年基于此设计了自然语言处理（Natural Language Processing, NLP）模型 Bert，旨在更好地理解人类每天使用的自然语言。2022 年，同样运用 Transformer，由 OpenAI 开发的 ChatGPT（Generative Pre-training Transformer）面向公众问世^[123-125]，其几乎可以谈论任何主题的能力，迅速引爆舆论。截至目前，深度学习还在迅速发展，应用场景正不断的渗透到各个领域。

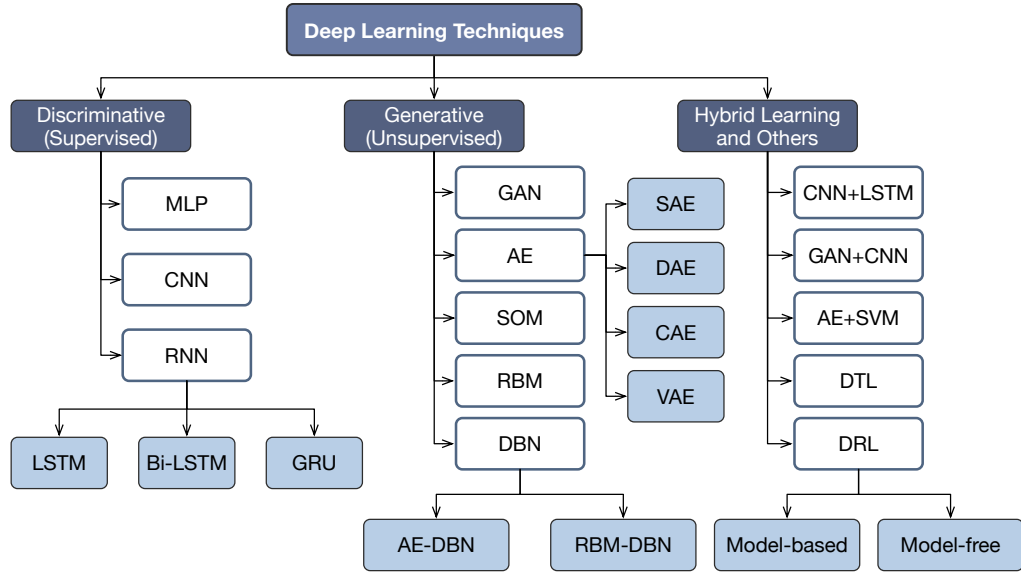
2.1.2.2 深度学习的基本概念

Sarker [126]按照传统机器学习的分类方式，将深度学习分为三类：（1）有监督或识别式学习（2）无监督或生成式学习（3）混合式学习及其他。图2-5展示了每一类中的代表型算法以及他们的变体。其中最经典的模型是多层感知机（Multi-layer Perceptron, MLP）。

感知机是神经网络中的一个神经元，是构成神经网络的最基本结构。

$$y = \text{sgn} \left(\sum_i^m w_i x_i + b \right). \quad (2-19)$$

公式2-19展示了感知机的工作原理：首先对输入端的数据 $\{x_1, x_2, \dots, x_m\}$ 赋予权


 图 2-5 深度学习分类^[126]。

重 $\{w_1, w_2, \dots, w_m\}$ ，然后对所有结果求和并加上偏置 b ，最后经过激活函数 sgn 的处理得到输出值 y 。其中权重 w 和偏置 b 为感知机待学习的参数。激活函数的主要作用是将输出结果非线性化，从而使神经网络可以拟合非线性关系。原始感知机的激活函数是一个阶跃函数 $\text{sgn}(x)$ ，当 $x \geq 0$ 时 $\text{sgn}(x) = 1$ ，当 $x < 0$ 时 $\text{sgn}(x) = -1$ 。除此以外，常用的激活函数还有 ReLU、sigmoid、tanh、softmax 等，它们有各自的特性及适用范围^[127]。

典型的多层感知机是一个全连接的前馈人工神经网络（Feedforward ANN），其特征是数据在每层之间单向流动，每个神经元的输入是上一层所有神经元的输出。图2-6展示了一个包含两个隐层的多层感知机模型结构。

单层感知机模型是线性模型加激活函数，因此与训练线性模型一样，梯度下降法可直接用于训练单层感知机模型^[116,128]。对于多层感知机也可用类似方法。以回归问题为例，假设样本 (x, y) 的输出为 $\hat{y}$ ， $y = (y^1, y^2, \dots, y^T)$ 。回归问题的输出层一般不需要激活函数，

$$\hat{y} = \sum_{h=1}^H w_{ih} z_h + b_i \quad (2-20)$$

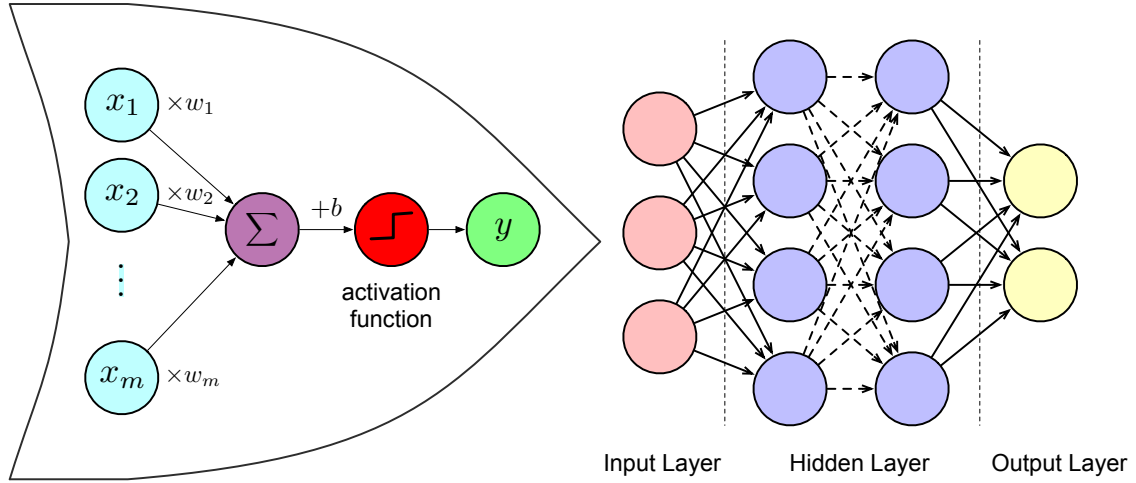


图 2-6 多层感知机模型、全连接神经网络结构示意图。

其中， z_h 为最后一层隐层的输出，共有 H 个输出。

$$z_h = \sigma \left(\sum_{j=1}^J w_{hj} z_j + b_j \right) \quad (2-21)$$

其中， σ 为激活函数， z_j 为上一层的输出，共有 J 个输出，以此类推。定义损失函数为方均误差，

$$L = \frac{1}{2} \sum_{t=1}^T (y^t - \hat{y}^t)^2 \quad (2-22)$$

要训练的参数有 $w_{ih}, w_{hj}, \dots$ 及 $b_i, b_j, \dots$ 。类似公式2-3，要更新参数，只需求出损失函数对相应参数的梯度即可。例如，由链式法则，误差对参数 w_{hj} 的梯度可以写为，

$$\frac{\partial L}{\partial w_{hj}} = \frac{\partial L}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{hj}}. \quad (2-23)$$

按照顺序逐层回溯即可得到所有参数的对应结果。由于误差由输出反向传递给了输入，所以这种方法称为反向传播法^[116]。反向传播是深度学习中最重要模型优化方法。

2.1.2.3 循环神经网络

在前馈网络中，数据单向流通，神经元自身、神经元之间都没有数据交换。与前馈神经网络不同，循环神经网络（Recurrent Neural Network, RNN）在隐层中添加了神经元的自连接或神经元间的互连接，使数据不再单向流通^[129]。这种结构可以使得神经元存储来自先前输入的信息，捕捉数据中的长期依赖关系。

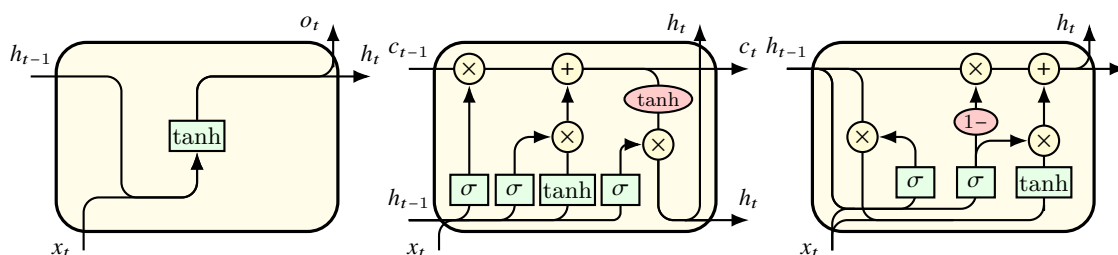


图 2-7 循环神经网络结构。由左至右分别为原始 RNN、LSTM 以及 GRU。

RNN 存在“梯度爆炸”或“梯度消失”的问题，导致原始的 RNN 很难进行长序列的学习。为了解决这些问题，人们提出了 RNN 的各种扩展和变体，如长短时记忆^[130]（LSTM）、门控递归单元^[131]（GRU）和注意机制^[122]（Attention）等。这些模型引入了门和记忆单元等附加组件，以控制信息流并增强 RNN 的学习能力。

2.1.2.4 卷积神经网络

卷积神经网络（Convolutional Neural Network, CNN）是一种专为处理图像、视频或文本等结构化数据设计的人工神经网络，广泛应用于计算机视觉（Computer Vision, CV）和自然语言处理（Natural Language Processing, NLP）领域，在图像分类、物体检测、人脸识别、情感分析、机器翻译和语音识别等许多视觉和语言任务中都具有良好的表现。

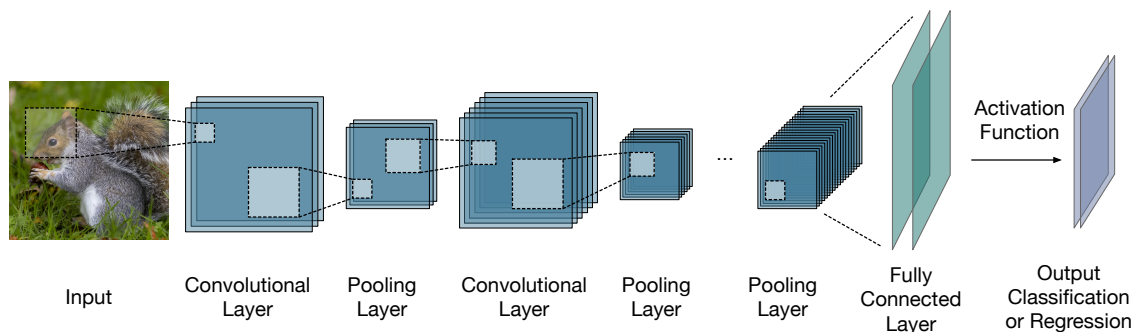


图 2-8 卷积神经网络典型结构。

图2-8展示了一个典型的卷积神经网络结构。卷积层（Convolutional Layer）是CNN的主要组成部分，它将一组过滤器应用于输入数据，以提取局部特征，例如图像的边缘、形状、纹理或文本中的单词等。卷积层之后通常是池化层（Pooling Layer），池化层通过对输入数据的某个区域应用最大值、平均值或求和等函数来降低数据的空间维度。池化层的作用是在卷积神经网络中对数据进行降维，提取更高层次的特征，增强模型的不变性和泛化能力。卷积层和池化层通常会重复多次，形成一个分层特征提取过程，其中每一层都能识别更复杂、更抽象的模式。CNN的最后一层通常是一个全连接层（Fully Connected Layer），根据任务的不同充当分类器或回归器的作用，并产生网络的最终输出结果。

2.1.2.5 生成式神经网络

生成式网络（Generateive network）是一类无监督深度学习方法。比较常见的算法有自编码器（Auto-encoder, AE），深度信念网络等。Goodfellow et al. [121]提出的生成对抗网络（Generative Adversarial Network, GAN）是生成式网络最具突破性的代表之一，通过加入对抗识别模块，显著减少了神经网络的使用层数，极大的提高了模型的训练效率。原始的GAN是一种无监督学习模型，但随着GAN的成功，很快便发展出一系列衍生和变体模型^[132-134]。

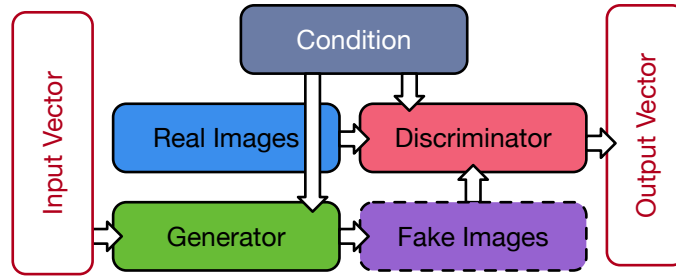


图 2-9 条件生成对抗网络结构。

图2-9 展示了条件生成对抗网络（CGAN）^[132] 的基本结构。GAN 主要由两个相互“对抗”的模块组成：生成器（Generator）与判别器（Discriminator）。生成器的作用是在给定一组随机向量 z 时，生成对目标图像 x 的模拟 $G(z)$ 。判别器的作用是区分生成器的输出与训练集中真实图像之间的差异，判断生成器的输出是否可以“以假乱真”。判别器的引入极大的提升了网络的训练效率。

GAN 的目标函数为，

$$\arg \min_G \max_D \mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_x [\log D(x)] + \mathbb{E}_z [\log(1 - D(G(z)))] . \quad (2-24)$$

其中, $\arg \min/\max f(x)$ 表示: 当 $f(x)$ 取最小/最大值时, 求解参数 x 的值。 $\mathbb{E}$ 表示数学期望。 $D(x)$ 表示输入为 x 时判别器的输出。对判别器, 需要调节判别器参数, 令 $\mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(G(z)))]$ 最大化 ($\max_D$)。对生成器, 需要在固定判别器参数时调节生成器参数, 令 $\mathbb{E}_z[\log(1 - D(G(z)))]$ 最小化 ($\min_G$)。在实际训练生成器时, 通常选择令 $\mathbb{E}_z[\log D(G(z))]$ 最大化。

如果对生成器和判别器提供额外的约束条件 y , 则可以将 GAN 扩展为条件生成对抗网络。此时, 目标函数变为

$$\arg \min_G \max_D \mathcal{L}_{CGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x|y)] + \mathbb{E}_{z,y}[\log(1 - D(G(z|y)))] . \quad (2-25)$$

2.1.3 强化学习

在人类或动物的行为模式中, 一个动作是否产生奖励会影响其下一步行动。强化学习 (Reinforcement Learning, RL) 试图模仿这种决策模式, 通过学习环境对主体行为的反馈优化主体行为, 从而获取更多奖励^[135-136]。

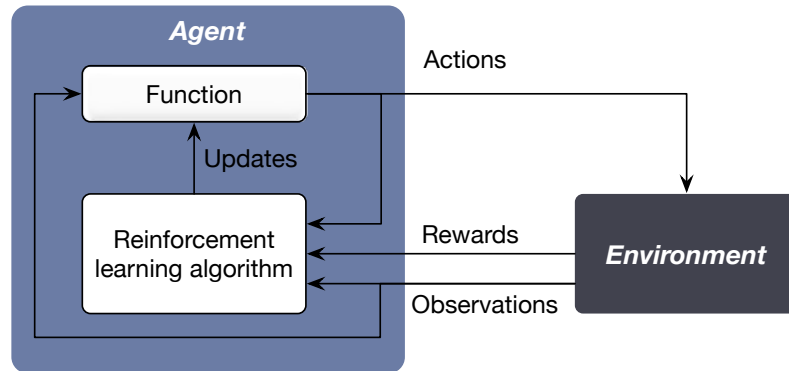


图 2-10 强化学习。

图2-11展示了强化学习算法的分类图。强化学习有无模型 (model-free) 和有模型 (model-based) 两种模式。无模型强化学习算法不会建立环境的模型, 而是直接从与环境的交互中学习, 有模型强化学习算法会首先尝试学习或给定一个环境模型, 然后使用这个模型来进行策略并进行学习。强化学习在各个领域都有应用, 包括机器人学、游戏玩法、金融和医疗保健。

2.1.3.1 马尔可夫决策过程

马尔可夫决策过程 (Markov Decision Processing, MDP) 是强化学习问题在数学上的理想化形式。一个马尔可夫决策过程由元组 (S, A, P, R, γ) 定义, 其中, (1) S

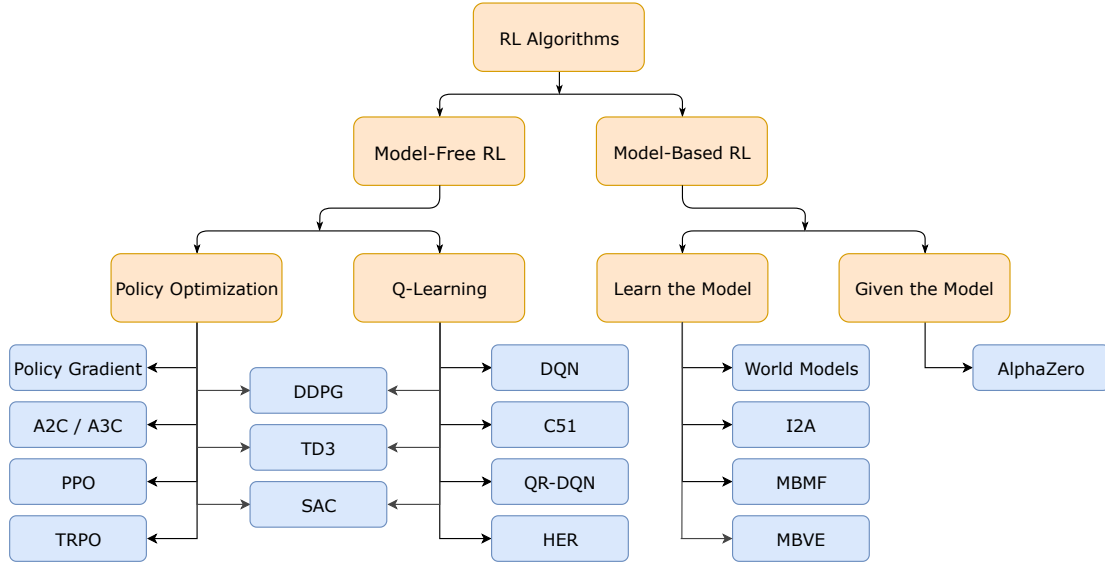


图 2-11 强化学习分类（图片来源：OpenAI）。

为状态集， $s \in S$ 表示环境中的一个特定情境或配置；(2) A 为动作集， $a \in A$ 表示动作集中可能采取的一个行动；(3) P 为状态转移概率函数， $P(s'|s, a)$ 表示通过采取行动 a 从状态 s 过渡到状态 s' 的概率；(4) R 为奖励函数， $R(s, a, s')$ 表示通过采取行动 a 从状态 s 过渡到状态 s' 后获得的即时奖励；(5) γ 为折损因子，是一个介于 0 和 1 之间的参数，确定未来奖励的重要性。折后的累积奖励为，

$$G = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots . \quad (2-26)$$

其中， R_i 表示第 i 期奖励。

MDP 的目标是找到一个策略（policy），使得累积奖励的期望最大。策略 π 是从状态到行动概率的映射， $\pi(a|s)$ 表示当状态为 s 时，采取行动 a 的概率。策略决定了智能体（agent）的决策行为。累积奖励通常通过值函数进行描述。当智能体遵循特定策略 π 时，在状态 s 下未来累积奖励的期望值称为策略 π 的状态值函数，

$$V^\pi(s) = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi[R_{t+k+1} | s_t = s] . \quad (2-27)$$

在状态 s 下采取行动 a 后未来累积奖励的期望值称为策略 π 的状态行动值函数，

$$Q^\pi(s, a) = \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi [R_{t+k+1} | s_t = s, a_t = a] . \quad (2-28)$$

策略、状态值函数与状态行动值函数满足如下关系，

$$V^\pi(s) = \sum_{a \in A} \pi(a|s) Q^\pi(s, a) . \quad (2-29)$$

贝尔曼方程 (Bellman Equation) 给出了某个状态值函数与其后继状态值函数间的关系，

$$V^\pi(s) = \mathbb{E}_\pi [R_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s] \quad (2-30)$$

最优价值函数可以用贝尔曼最优方程进行递归性描述，

$$V^*(s) = \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^*(s')] \quad (2-31)$$

类似的也可以写出状态行动值函数 $Q(s, a)$ 对应的贝尔曼方程。寻找最优策略的问题可以转换为寻找最优的状态值函数或状态行动值函数。

求解最优值函数最常用的方法有动态规划 (Dynamic Programming, DP)、蒙特卡洛 (Monte-Carlo, MC) 和时序差分学习 (Temporal Difference Learning, TDL) 等。

2.1.3.2 动态规划

动态规划直接根据贝尔曼公式将问题分解成一步步的子问题，然后进行递归求解。这就要求可以对环境的动态变化进行建模，确定状态转移概率函数和奖励函数的具体形式 (model-based)。但是在现实问题中，往往无法对问题进行如此完备的描述。此外，动态规划的计算复杂度较高，还有可能遭遇由于状态数量激增导致的“维度灾难”。

2.1.3.3 蒙特卡洛

蒙特卡洛方法不需要对问题进行完备的描述，直接从经验中学习规律 (model-free)。选取一个样本使其遍历整个决策过程，可以得到该样本对应的状态、行动、

以及奖励序列，这个过程称为一幕（episode）。蒙特卡洛法通过随机采样法，对不同样本的幕进行平均，可以更高效的处理动态规划的问题。蒙特卡洛可以分为两种算法：同轨策略（on-policy）法和离轨策略（off-policy）法。同轨策略法直接使用要学习的目标策略来生成样本并更新值函数及策略；离轨策略法则将策略分为了目标策略与行动策略，行动策略专门用来生成样本供目标策略进行学习。蒙特卡洛法需要令每个样本都遍历整个决策过程，即到达幕的末尾，这可能导致收敛过慢甚至难以收敛。此外，蒙特卡洛结果的方差往往较大。

2.1.3.4 时序差分学习

第三种方法是时序差分学习，它结合了动态规划与蒙特卡洛。时序差分不需要令每个样本都遍历整个决策过程，它通过学习样本当前状态与后继状态间的差别来优化值函数，因此极大的提升了蒙特卡洛方法的学习效率。

SARSA 与 Q-learning 分别是同轨策略与离轨策略下，时序差分算法的典型代表。SARSA（state-action-reward-state-action）是同轨策略下的时序差分，

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2-32)$$

Q-learning 是一种离轨策略下的时序差分学习^[137]。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (2-33)$$

2.1.3.5 函数逼近法

函数逼近法的基本思想是用一个参数化的函数来近似值函数或策略函数，然后通过学习调整参数来优化性能。函数逼近法可以分为线性函数逼近和非线性函数逼近，其中非线性函数逼近又可以进一步分为基于表格的方法和基于神经网络的方法。

深度强化学习（Deep Reinforcement Learning, DRL）是利用深度神经网络来逼近值函数和策略函数的一类学习方法。由此诞生了许多在复杂环境中表现出色的算法，如 Deep Q-Networks^[138]（DQN）和 Proximal Policy Optimization^[139]（PPO）等。

2.1.4 集成学习

集成学习 (ensemble learning/methods) 是指通过结合多个学习器构建的一类学习方法^[140-142]。Dietterich [143]指出了集成学习可能优于单一学习器的三点根本原因：(1) 统计学原因：单一学习器在数据集不足时可能产生拥有相同精度的不同结果。通过集成这些单一学习器，可以在统计学上降低错误答案出现的概率。(2) 计算型原因：很多算法在寻找最优解时会卡在局部最优解而非全局最优解，例如深度学习中的梯度下降和决策树中的贪心分裂 (Greedy Splitting)，都属于 NP-hard 问题。通过集成初始位置不同的单一学习器，集成学习可以更加接近全局最优解。(3) 表征型原因：想要学习的真实结果可能存在于未知的假设区域，不能被单一学习器所表征。集成不同的算法可以扩大他们的表征范围，从而找到更真实的结果。

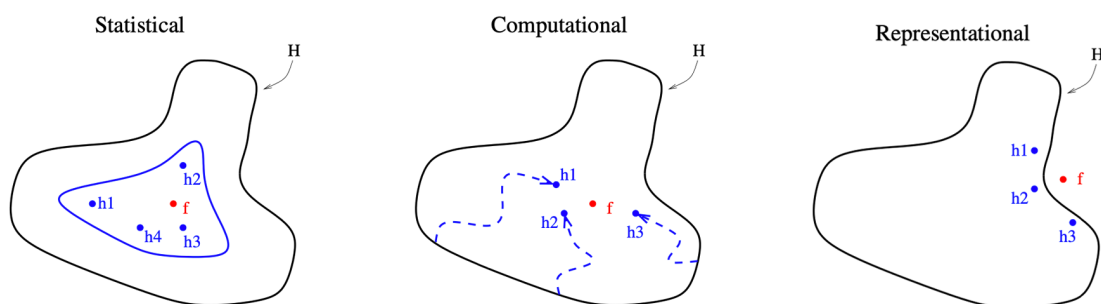


图 2-12 集成学习优于单一学习器的三点根本原因。

图2-12形象的展示了上述三点原因。外围黑色实线表示单一学习器所能表征的假设空间的范围， $h_1, h_2, h_3, \dots$ 分别表示单一学习器给出的结果， f 代表真实结果。相比单一的学习器，集成学习可以显著提升模型的精度、降低模型的偏差。

常用的集成学习主要有三种结构：Bagging、Boosting 及 stacking。

2.1.4.1 Bagging

Bagging(Bootstrap aggregating) 通过 Bootstrap 对数据进行重采样，然后用独立的模型学习这些样本，最后集成所有习得的模型^[140,142]。随机森林 (Random Forest) 是 Bagging 算法的典型代表^[144-145]。随机森林使用决策树作为学习器，将所有学习器在 bootstrap 样本上习得结果的平均值作为最终结果。

2.1.4.2 Boosting

Boosting 使用弱学习器 (例如深度较浅的决策树) 构建一个学习器序列，令序列中的每个学习器学习上一个学习器的输出，最后集成所有弱学习器得到一个

强学习器。Boosting 算法的代表是 AdaBoost^[146] (Adaptive Boosting) 和 Gradient Boosting^[147-148]。

梯度提升 (Gradient Boosting) 是近年表现最突出的机器学习算法之一。其主要思想是利用一个弱学习器去学习上一个弱学习器损失函数的负梯度 (Gradient)。常用的 Gradient Boosting 框架有华人学者陈天奇开发的 XGBoost^[149]，微软的 LightGBM²，以及俄罗斯团队开发的 CatBoost³。

2.1.4.3 Stacking

Stacking 采取的集成方式是在相同数据集上使用不同的模型进行学习，再将学习到的结果构建成新的数据集，然后利用另一个模型，称为元学习器或二级学习器，进行最终的学习^[150]。

2.2 机器学习在算法交易中的应用

由于机器学习在处理复杂非线性问题上具有明显优势。越来越多的分析师及研究人员开始使用人工智能技术分析庞大的金融数据集，识别复杂的模式，并开发用于交易和投资组合管理的模型。

2.2.1 订单簿模拟

订单簿模拟是量化交易中一个非常重要的工具，它帮助交易者更深入地理解市场动态，测试和优化交易策略，同时管理潜在的风险^[151-155]。订单簿模拟最简单的方法是历史数据回放，这种方法可以帮助交易者理解过去特定事件对市场的影响，但是很难反应真实交易环境中的各种复杂环境变化。除此以外，还有多种方法和技术被用于提高模拟的真实性和有效性。

- 蒙特卡洛模拟。蒙特卡洛模拟方法通过随机过程生成数据，从而模拟可能的市场情况。这种方法特别适用于评估在不确定的市场条件下交易策略的表现。蒙特卡洛模拟可以提供对潜在市场行为的广泛洞察，帮助交易者准备应对各种可能的市场波动。
- 基于代理的模型^[156-157]。基于代理的模型涉及创建一个虚拟的市场环境，其中多个交易代理 (Agent) 根据预设策略进行交易。这种模型有助于分析和理

¹ XGBoost: <https://xgboost.ai>

² LightGBM: <https://lightgbm.readthedocs.io/>

³ CatBoost: <https://catboost.ai>

解市场中的复杂交互作用。每个代理代表了一个市场参与者，他们的交易决策和行为模式可以基于实际市场数据或假设的行为模式来设定。

- 事件驱动模拟。事件驱动模拟考虑到了市场中的关键事件，如重大新闻发布、政策变动、经济数据公布等。这些事件对市场的影响通常是显著的，因此，在模拟中加入这些因素可以帮助交易者理解和预测这些事件对市场可能产生的影响。
- 机器学习和人工智能。随着技术的进步，机器学习和人工智能在订单簿模拟中扮演着越来越重要的角色。利用这些先进技术，可以预测和模拟市场行为的复杂模式，甚至在某些情况下，可以模拟市场的未来动向。深度学习等方法能够处理和分析大量的历史数据，从而生成更精确的市场模拟结果。例如，Takahashi et al. [158], Li et al. [159], Coletta et al. [160], Cont et al. [161] 使用生成对抗网络模拟了订单簿，Briola et al. [162]使用 LSTM 深度网络模拟订单簿等。

2.2.2 价格预测

股票价格预测是金量化交易和算法交易研究的核心问题。通过分析历史数据、市场趋势和其他相关因素来预测未来的股票价格，对于投资者制定投资策略、管理风险和寻求收益都至关重要。虽然股票价格预测具有极高的价值，但其复杂性和不确定性意味着没有任何一种方法可以保证完全准确。投资者需要结合多种方法和工具，同时考虑市场的动态变化，灵活调整投资策略。

Main Algorithm	Studies
Neural Network	Zhang et al. [99], Rajihy et al. [163], Hu et al. [164], Shen et al. [165]
GAN	Zhang et al. [166], Diqi et al. [167]
CNN	Tsantekidis et al. [168], Gunduz et al. [169], Hoseinzade et al. [170], Cao et al. [171], Lu et al. [172], Ishwarappa et al. [173]
LSTM	Bao et al. [174], Minh et al. [175], Baek et al. [176], Shen et al. [177], Li et al. [178], Lee et al. [179]
SVM	Lee [180], Hu et al. [181], Rustam et al. [182], Liagkouras et al. [183], Yuan et al. [184], Mahmoodi et al. [185]
Decision Tree	Lai et al. [186], Zhou et al. [187], Carta et al. [188]
XGBoost	Jiang et al. [189], Kim et al. [190], Yun et al. [191], Almaafi et al. [192]
Reinforcement Learning	Huang [193], Pricope [194]

表 2-1 利用不同机器学习技术预测股票价格的相关研究摘录。

传统统计方法利用如移动平均线、指数平滑等来预测股价未来趋势，但随着人

工智能技术的发展，机器学习，特别是深度学习，在股票价格预测中的应用越来越广泛。表2-1 展示了近年一些利用不同机器学习、深度学习技术预测股票价格的研究工作。更多研究可以参考综述文章 Zou et al. [195], Kumbure et al. [196], Patel et al. [197]。

虽然许多研究表明机器学习可以有效提升股票价格预测的精度，但股票价格预测仍然面临着许多挑战和问题。首先，证券市场非常复杂，受多种因素影响，包括政治事件、经济变化、公司业绩等，这些因素往往难以准确量化。其次，市场参与者的行为模式本身就是非线性和动态的，这使得模型构建变得更加困难。此外，过拟合是机器学习面临的一个常见问题，特别是在使用高度复杂的模型时，市场不确定性带来的影响一直是一个重要的挑战。

2.2.3 情绪分析

金融情绪分析（Financial Sentiment Analysis）是预测资产价格走势的另一种重要手段。利用机器学习进行金融情绪分析一直面临两个难点：一是处理不同语言的特异性，一是缺少有标签的数据集。

传统机器学习的特征提取方式通常是对反应情绪的关键词进行计数^[198]，但这种方式的效率和准确度较低。深度学习通过词汇嵌入（word embedding）将文本映射为向量，然后交给神经网络进行学习。例如，Swathi et al. [101], Severyn et al. [199]通过分析 twitter 文本，结合深度学习算法对市场情绪进行了分析，并预测了市场走向。随着近年自然语言处理（NLP）取得的突破性进展^[200]，利用预训练模型解决上述问题开始变得可行。FinBert^[201] 第一次将 Bert 预训练模型运用于金融情绪分析并取得了一些突破进展。Lopez-Lira et al. [202]研究了利用 ChatGPT 进行市场情绪预测的可行性。

2.2.4 合规检测

股市合规检测是一个复杂的过程，旨在确保金融机构和个人参与者遵守相关法律、规则 and 标准。对算法交易影响较大的主要是市场操纵和市场欺诈行为。

股市操纵通常涉及通过虚假信息、误导性行为或其他手段人为影响股票价格或交易量，从而使市场参与者作出基于误导的决策。例如，通过内部交易、回购或其他手段创造虚假的市场活动，误导其他投资者；故意推高或压低股价，以便在不正当的价格水平买入或卖出；通过媒体、社交网络或其他渠道散播虚假或误导性信息，以影响股价等。Chullamonthon et al. [203]通过深度学习检测股价是否存在操纵。

股市欺诈涉及使用欺骗手段来误导投资者，通常是通过虚假声明、隐瞒重要信息或其他欺诈行为。例如，企业通过夸大收入、隐藏债务或其他方式操纵财务报表，误导投资者和市场；通过虚假的投资机会欺骗投资者等。Craja et al. [204]利用深度学习检测财务报表造假行为。Ali et al. [205]回顾了基于机器学习的市场欺诈行为检测的相关工作。

第三章 基于机器学习的高频算法交易研究

随着金融市场的快速发展和技术进步，高频算法交易在金融领域的应用越来越广。机器学习技术的发展为高频交易提供了新的机遇与挑战。通过结合多种机器学习手段，作者开发了一套基于 A 股股票市场的算法交易策略。本章将介绍策略的研究方法以及取得的成果。

3.1 研究背景

如前所述，近年来，在先进技术迅速融入交易实践的推动下，金融市场结构经历了一场变革性的演变。在这些技术进步中，高频算法交易已成为一股主导力量，重塑了传统模式，为市场运作带来了前所未有的效率和速度。与此同时，高频交易与机器学习的交集也引起了广泛关注，这为通过利用复杂的计算模型来增强交易策略提供了一条大有可为的途径。

在金融市场不断向着更加动态和复杂的方向发展的同时，对于能够自适应并具备智能化特征的交易算法的需求愈发显著。虽然传统的交易方法在某些情况下依然有效，但它们通常无法精准捕捉到市场条件快速变化的细微差异。因此，越来越多的人认识到，将高频交易与机器学习相结合，可能会为交易策略的高效实施开辟全新的维度。鉴于此，本文旨在深入探讨高频算法交易领域，特别是将机器学习方法应用于交易策略的开发和评估。

为了实现这一目标，研究团队采纳多种方法，开发出了一套能够实时学习并适应市场动态的算法交易策略。本章将首先介绍了典型的交易系统结构，之后介绍了研究的主要内容，包含原始数据的处理、特征工程、模型构建等，最后展示了模型的训练结果。

3.2 交易系统

图3-1显示了本文使用的交易系统结构，包含系统内不同组件间的交互和数据传输方向。以下是各部分的详细介绍：

- 交易所/数据服务器 (Exchange/Data Server)：这是交易系统连接到金融交易所或其他市场数据提供商的地方。它接收实时市场数据，包括：市场数据快照 (Market Data)、订单队列 (Order Queue) 和交易队列 (Transaction Queue)。
- 经纪人/客户端 (Brokers/Clients)：这些是经纪人、券商或其他使用交易系统

下订单的客户。他们可以将交易系统的交易指令转发给交易所。

- **网关 (Gateway)**: 网关是内部系统与外部服务器 (如交易所) 之间的通信点。进口网关 (**Gateway In**) 处理来自不同数据提供商的交易数据, 如市场快照和交易信息等, 并将其传输到后续数据处理模块; 出口网关负责将订单管理器的订单发送至不同经销商或客户端。
- **历史市场数据库 (History Market Database)**: 该数据库用于存储历史市场数据, 用于研究和回测交易策略, 以确保策略在实时交易中的有效性。
- **订单簿构建器 (Book Builder)**: 该组件处理接收到的市场数据, 用以重构订单簿, 并将数据进行落地, 以及在实盘交易中将处理过的市场数据播放给策略模块。
- **订单管理器 (Order Manager)**: 它负责管理订单的生命周期, 包括订单的生成、发送、执行、确认和取消。**Order manager** 的功能有以下几点: 与交易平台或经纪商进行通信, 将订单发送到市场, 并接收订单的状态和成交信息; 与策略引擎进行交互, 根据策略的指令生成订单, 并将订单的执行情况反馈给策略引擎; 与风险管理模块进行协调, 检查订单是否符合风险控制的要求, 例如最大持仓量、最大下单量、最大损失等; 与数据库或文件系统进行存储, 记录订单的详细信息, 例如订单号、时间、价格、数量、方向、状态、成交量、成交价等。
- **SQL 数据库**: 这是一个结构化数据库, 存储订单管理器生成的有关订单、交易和可能的其他运营数据的信息。
- **研究/回测**: 系统的这一部分使用历史数据来测试交易策略在过去的表现。这对于开发新策略和完善现有策略至关重要, 是算法交易策略研究的核心内容。
- **策略 (Strategies)**: 这是算法交易策略模块, 用于分析接收到的数据, 并根据预定义的标准生成订单的交易信号。
- **风险控制 (Risk Control)**: 该组件实时监控交易策略的风险敞口。它确保策略符合风险参数, 并在必要时发送信号停止交易。
- **篮子管理器 (Basket Manager)**: 该模块管理一系列 (或“篮子”) 的订单, 并对它们进行分组以管理和执行策略。
- **用户界面 (User Interface)**: 这是用户与系统交互的前端, 用户可以通过它提交篮子, 接收并查看订单状态、实时市场数据并管理交易策略。
- **网页或系统监控/客户端**: 这可以是一个监控工具或客户端应用程序, 为系统管理员或用户提供查看系统性能和运营指标的界面。

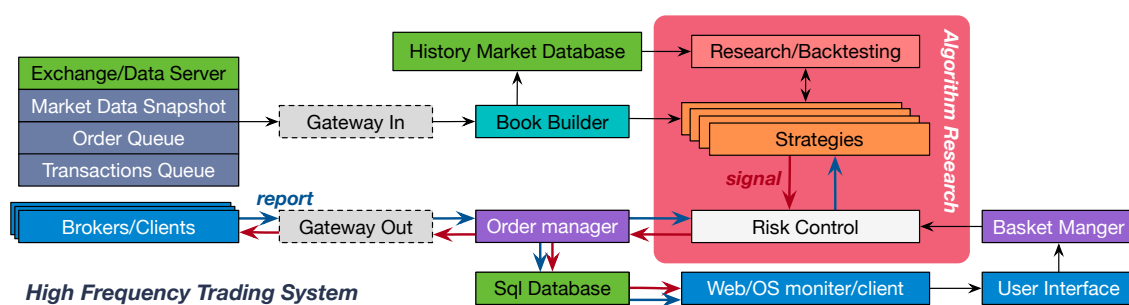


图 3-1 高频交易系统结构。

交易系统的工作流程通常从实时市场数据通过网关进入系统开始。然后，数据由订单簿构建模块处理，并被策略组件用来做出交易决策。最后，由策略生成的订单通过订单管理器，通过网关出与交易所通信。全过程中，风险持续被监控，所有操作都记录在数据库中。用户界面允许客户与系统互动，用于提交篮子、进行交易监控。

3.3 研究方法

3.3.1 策略的定义

假设交易任务要求在交易周期 $[0, T]$ 内交易数量 S 的某资产，那么，一个交易策略就是一个映射关系 $s_t = f(t)$ ，其中， s_t 为任意时刻 $t \in [0, T]$ 需要的买入量，且满足 $\int_0^T s_t = S$ 。对于实际交易中的离散情况，策略变为了两个序列：一个是时间序列 $\{t_1, t_2, \dots, t_n\}$ ，另一个是对应的交易量序列 $\{s_1, s_2, \dots, s_n\}$ ，其中， $0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq T$ ，且有 $\sum_{i=1}^n s_i = S$ 。如果每次交易的价格为 $\{p_1, p_2, \dots, p_n\}$ ，那么策略的目标是 $\max_{ask} \min_{bid} \sum s_i p_i$ ，即在买入 (bid) 时最小化交易额，卖出 (ask) 时最大化交易额。

传统交易策略，如 TWAP、VWAP 等，在交易时间得选取上都采取等时划分，即将交易时间 $[0, T]$ 均匀划分为 $n + 1$ 。

3.3.2 数据处理

第一章中已经介绍了 A 股数据的组织形式，本文的数据源来自万得宏汇 TDF 行情接口提供的 Level-2 数据，主要使用了股票的行情快照、逐笔成交、以及逐笔委托三种数据类型，具体数据结构见表 3-1。

Variable	Data Type	Description	Variable	Data Type	Description
行情快照			逐笔成交		
gid	int	本日编号	gid	int	本日编号
nTime	int	时间 ^a	nItems	int	数据个数
nStatus	int	状态 (见附表A-1)	nTime	int	成交时间 ^a
nPreClose	unsigned int	前收盘价	nIndex	int	成交编号
nOpen	unsigned int	开盘价	nPrice	int	成交价格
nHigh	unsigned int	最高价	nVolume	int	成交数量
nLow	unsigned int	最低价	nTurnover	int	成交金额
nMatch	unsigned int	最新价	chBSFlag	char	买卖方向 ^b
nAskPrice[10]	unsigned int	申卖价	chOrderKind	char	成交类型
nAskVol[10]	unsigned int	申卖量	chFunctionCode	char	成交代码
nBidPrice[10]	unsigned int	申买价	chResv	char	保留
nBidVol[10]	unsigned int	申买量	nAskOrder	int	卖方委托序号
nNumTrades	unsigned int	成交笔数	nBidOrder	int	买方委托序号
iVolume	__int64	成交总量	逐笔委托		
iTurnover	__int64	成交总金额	gid	int	本日编号
nTotalBidVol	__int64	委托买入总量	nItems	int	数据个数
nTotalAskVol	__int64	委托卖出总量	nTime	int	委托时间 ^a
nWeightedAvgBidPrice	unsigned int	加权平均委买价格	nIndex	int	委托编号
nWeightedAvgAskPrice	unsigned int	加权平均委卖价格	nPrice	int	委托价格
nIOPV	int	IOPV 净值估值	nVolume	int	委托数量
nYieldToMaturity	int	到期收益率	chType	char	委托类别
nHighLimited	unsigned int	涨停价	chBSFlag	char	委托买卖类型 ^c
nLowLimited	unsigned int	跌停价	chResv	char	保留
chPrefix[4]	char	证券信息前缀			

^a 格式 HHMMSSmmmm(10 毫秒)

^b 买: 'B', 卖: 'S', 不明: ''

^c 买: 'B', 卖: 'S', 撤单: 'C'

表 3-1 行情快照、逐笔成交、逐笔委托数据结构。

3.3.2.1 订单簿重构

A 股行情快照数据每 3 秒更新一次, 通过订单簿重构可以得到更细部的订单簿变化, 提供更灵活且高效的算法开发环境。

深交所的重构相对直接, **Order** 与 **Transaction** 之间理论上所有 ID 都一一对应。但是根据上交所交易规则, 在连续竞价阶段, 若一笔订单被一次性全部撮合, 则不会发布该订单的剩余委托数量。这意味着对于市价单, 有可能收不到其原始的 **Order** 信息。此时需要综合考虑订单的到达顺序等, 对订单簿进行重构。本文使用更新频率为 1 秒的重构订单簿数据进行相关研究。

3.3.2.2 噪音过滤

高频交易产生的大量数据受到市场微观结构和交易环境（如延迟、丢包）等因素的影响，会产生噪声。这些噪声会扭曲数据中的真实信号，如果不进行处理，会进一步影响模型的准确性，导致错误的交易信号。通过去噪，有助于提高交易模型的性能，降低风险，并促进更有效的市场运作。

移动平均 (MV) 是最简单的低通噪音过滤手段，其原理是对一定时间段内的数据进行算术平均，然后将这个平均值作为该时间段的代表值：

$$\hat{p}_i = \frac{1}{w+1} \sum_{n=i-w}^i p_n \quad (3-1)$$

移动平均的优点是简单易用，能够有效地去除随机噪声，但是对于数据的变化反应较慢，而且会造成一定的滞后效应。

指数移动平均 (EMV) 的原理是对最近的数据赋予更高的权重，对较早的数据赋予较低的权重，然后用加权平均的方法来计算平均值：

$$\hat{p}_i = \alpha p_i + (1 - \alpha) \hat{p}_{i-1} \quad (3-2)$$

其中， α 是确定指数移动平均权重的参数。可以看到 EMV 权重按照指数函数递减，因此又称为指数衰减平均。指数移动平均对于数据的变化反应较快，能够更好地跟踪数据的趋势，而且滞后效应较小，但是对于随机噪声的抑制能力较弱，而且需要确定合适的权重参数。

小波变换去噪是一种利用小波分解和阈值处理的方法，可以有效地消除信号中的噪声成分，保留信号的有用信息。连续小波变换 (Continuous Wavelet Transform, CWT) 的形式为，

$$WT(a, \tau) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) * \psi\left(\frac{t-\tau}{a}\right) dt. \quad (3-3)$$

其中， a 为尺度，控制变换的宽度， τ 为平移量。

小波变换去噪的基本步骤如下：首先，对信号进行小波分解，得到不同尺度和频率的小波系数。其次，根据小波系数的幅值大小，选择一个合适的阈值，将小于阈值的小波系数置为零或进行收缩，认为这些系数主要是由噪声引起的。最后，对处理后的小波系数进行小波重构，得到去噪后的信号。

Sun et al. [206]展示了小波变换处理高频噪音过滤的能力。本文将研究小波变

换在 A 股高频数据中的去噪表现，并与传统去噪方法进行比较。小波变换去噪的效果和性能取决于多个因素，如小波基的选择，分解层数的确定，阈值函数的设计，阈值的计算等。不同的选择和方法可能适用于不同的信号和噪声类型，因此需要根据具体的问题和目的来进行调整和优化。

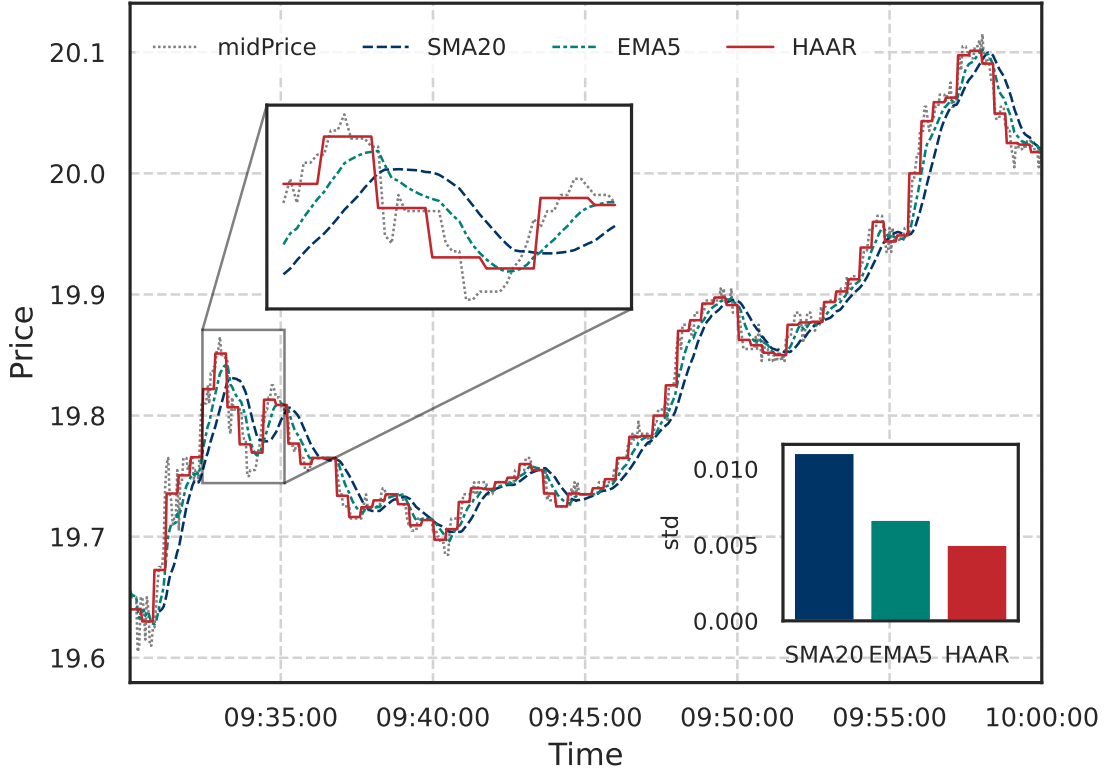


图 3-2 不同方法的噪音过滤对比。

本文使用 HAAR 小波变换，其基函数为，

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leq t < \frac{1}{2} \\ -1, & \text{if } \frac{1}{2} \leq t < 1 \\ 0, & \text{otherwise} \end{cases} \quad (3-4)$$

图3-2展示了分别使用移动平均、指数移动平均、以及 HAAR 小波变换，对股票价格噪音的处理结果。子图中展示了真实价格与降噪后价格的标准差。可以看到，小波变换不仅有效的去除了高频噪音，而且保留了更多的细部结构。

3.3.2.3 因子计算与特征工程

本文利用 A 股 Level-2 数据构造描述股票一分钟收益率的特征，每个特征定义为一个因子 `alpha`。

```

1  quant_alpha_s001::quant_alpha_s001(const quant::XmlNode& _cfg)
2      : quant_base_alpha(_cfg), didxNow(-1) {
3      dpi = quant_data_api::getInstance(); // dpi
4      univ = dpi->get_univ();
5      tkrs = univ->get_tkrs();
6      auto size = tkrs.size();
7
8      ntick = cfg.GetAttrDefault("ntick", 20);
9      alpha = new float[size]; // alpha
10     memset(alpha, 0xff, sizeof(float) * size);
11     ts_mean_ask_bid_spread = new TsMean<float, float>*[size];
12     for (size_t i = 0; i < size; ++i) {
13         ts_mean_ask_bid_spread[i] = new TsMean<float, float>(ntick);
14     }
15 }
16
17 quant_alpha_s001::~~quant_alpha_s001() {
18     if (alpha) delete[] alpha;
19     if (ts_mean_ask_bid_spread) delete[] ts_mean_ask_bid_spread;
20 }

```

图 3-3 因子构造函数和析构函数样例。

图3-3展示了本文使用的因子构造函数和析构函数的代码样例。构造函数 `quant_alpha_s001::quant_alpha_s001` 接受一个类型为 `quant::XmlNode` 的配置文件 `_cfg`，用于初始化因子构造函数。通过成员初始化列表（Member Initialization List）对基类 `quant_base_alpha` 进行初始化，同时对成员变量 `didxNow` 进行初始化为 -1。构造函数主要包含以下几个步骤：(1) 获取数据接口实例 `dpi`，获取该实例的股票池（universe）并存储到成员变量 `univ` 中，通过 `univ` 获取股票池中的所有股票代码并存储到成员变量 `tkrs` 中。获取股票池的大小并存储到变量 `size` 中。(2) 初始化成员变量，初始化 `ntick` 为配置文件中属性 `ntick` 的值，默认值为 20。动态分配一个大小为 `size` 的浮点型数组 `alpha`，并使用 `memset` 将数组中的所有元素初始化为 `0xff`，用于存储因子的计算结果。(3) 时序均值对象的动态分配，动态分配一个指针数组 `ts_mean_ask_bid_spread`，每个元素都是一个指向 `TsMean<float, float>` 类型对象的指针，用于存储因子计算的中间变量。使用 `for` 循环为每个股票代码创建一个 `TsMean<float, float>` 对象，该对象用于计算时序均值，其中参数 `ntick` 表示均值的时间窗口大小。析构函数 `quant_alpha_s001::~~quant_alpha_s001` 用于释放动态分配的内存，防止内存泄漏。在析构函数中使用 `delete[]`，检查并释放 `alpha` 数组和 `ts_mean_ask_bid_spread` 数组。

```

1  bool quant_alpha_s001::push_interval_data(long didx, long tidx) {
2      // Update pool
3      if (didxNow != didx) {
4          didxNow = didx;
5          pool = dpi->get_pool(didx);
6          vector<long> dtes = dpi->get_dlyTimes();
7          cerr << "didxNow = " << didxNow << " " << dtes[didxNow] / 1000000 << " "
8               << current() << endl;
9      }
10
11     // Update mMktData
12     ret = pool->get_data(tidx);
13     tie(cidx, idx, type, val) = ret;
14
15     if (type == TYPE_MD) {
16         auto md = val.md;
17         if (md->nAskVol[0] > 0 && md->nBidVol[0] > 0) {
18             auto spread = 1000.0f * (md->nAskPrice[0] - md->nBidPrice[0]) /
19                             (md->nAskPrice[0] + md->nBidPrice[0]);
20             ts_mean_ask_bid_spread[cidx]->onNew(spread);
21             alpha[cidx] = ts_mean_ask_bid_spread[cidx]->getValue();
22         } else {
23             alpha[cidx] = std::numeric_limits<float>::quiet_NaN();
24         }
25     }
26     // cerr << "quant_alpha_s001::push_interval_data end" << tidx << endl;
27
28     return true;
29 }
30
31 float* quant_alpha_s001::get_forecasts() { return alpha; }

```

图 3-4 因子计算样例。

图3-4展示了因子计算的样例代码，主要用于处理时序数据，并更新模型中的 `alpha` 值。`push_interval_data` 函数用于推送数据、更新股票池和计算 `alpha` 值。该函数主要包含以下几个步骤：(1) 检查当前的日期索引 `didxNow` 是否与数据的日期索引 `didx` 不同，如果不同，则更新股票池 `pool` 和 `didxNow`。输出一些调试信息，包括当前的日期索引 `didxNow`、对应的日期信息、以及当前时间。(2) 通过调用 `pool->get_data(tidx)` 获取数据，其中 `tidx` 是股票 `tick` 索引。解包 `ret`，其中包含 `cidx` (股票代码索引)、`idx` (数据索引)、`type` (数据类型) 和 `val` (数据值)。如果数据类型 `type` 为 `TYPE_MD` (市场快照数据) 时，则提取市场数据对象 `md`。(3) 判断卖一和买一的成交量是否大于零，如果是，则计算买卖价差 (`spread`) 的百分比。调用 `ts_mean_ask_bid_spread[cidx]->onNew(spread)` 更新时序均值对象。将计算得到的时序均值赋值给 `alpha[cidx]`。如果卖一或买一的成交量不大于零，将 `alpha[cidx]` 设置为浮点数的 `NaN`。(4) 最后，函数返回 `true`，表示处理成功。`get_forecasts` 函数用于获取计算得到的 `alpha` 数组，直接返回成员变量

alpha 的指针。

编号	简单描述	数据	数据周期	时间复杂度
A0	最近四分钟与最近两分钟之间买卖双方成交量总和的比值	逐笔成交	最近四分钟	$O(1)$
A1	当前逐笔成交价格与最近四分钟逐笔成交 VWAP 的比值 (价格突破)	逐笔成交	最近四分钟	$O(1)$
A2	最近两分钟逐笔成交买卖双方时间分布的不均衡性 (估算前卖后买还是前买后卖)	逐笔成交	最近两分钟	$O(1)$
A3	最近 30 秒逐笔成交价格涨幅	逐笔成交	最近 30 秒	$O(1)$
A4	最近 30 秒逐笔委托买卖双方时间分布的不均衡性 (估算前卖后买还是前买后卖)	逐笔委托	最近 30 秒	$O(1)$
A5	最近两分钟逐笔成交买卖双方时间分布的不均衡性 (估算前卖后买还是前买后卖, 注意和 00003 的区别)	逐笔成交	最近两分钟	$O(1)$
A6	单前时刻订单簿卖方最小委托量与卖方五档总量的比值	行情快照	当前时刻	$O(1)$
A7	当前时刻订单簿一档委比	行情快照	当前时刻	$O(1)$
A8	最近一分钟之内订单簿买卖双方 10 档委托总额的比值	行情快照	最近一分钟	$O(1)$
A9	最近 30 秒内行情快照买方 10 档委托额的最大值和最小值之间的时间差值	行情快照	最近 30 秒	$O(N)$

表 3-2 因子样例。

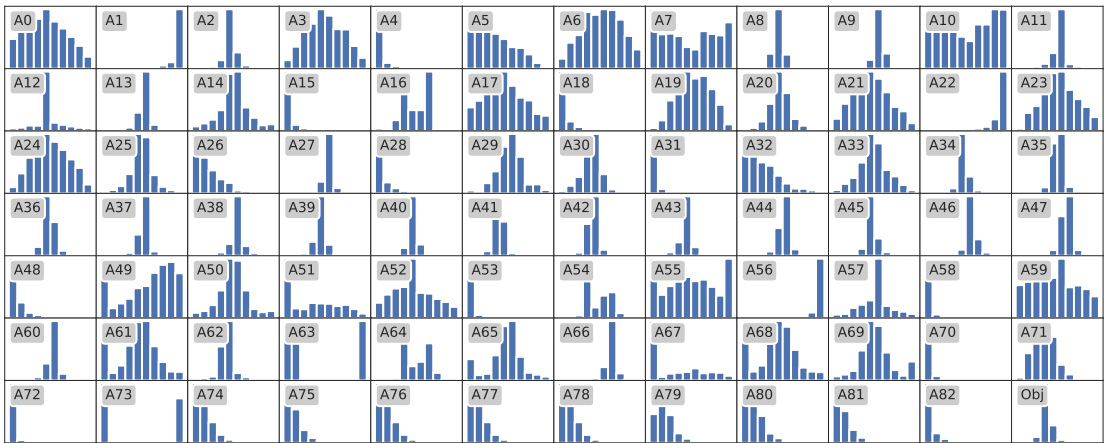


图 3-5 各因子与一分钟收益率的分布直方图。

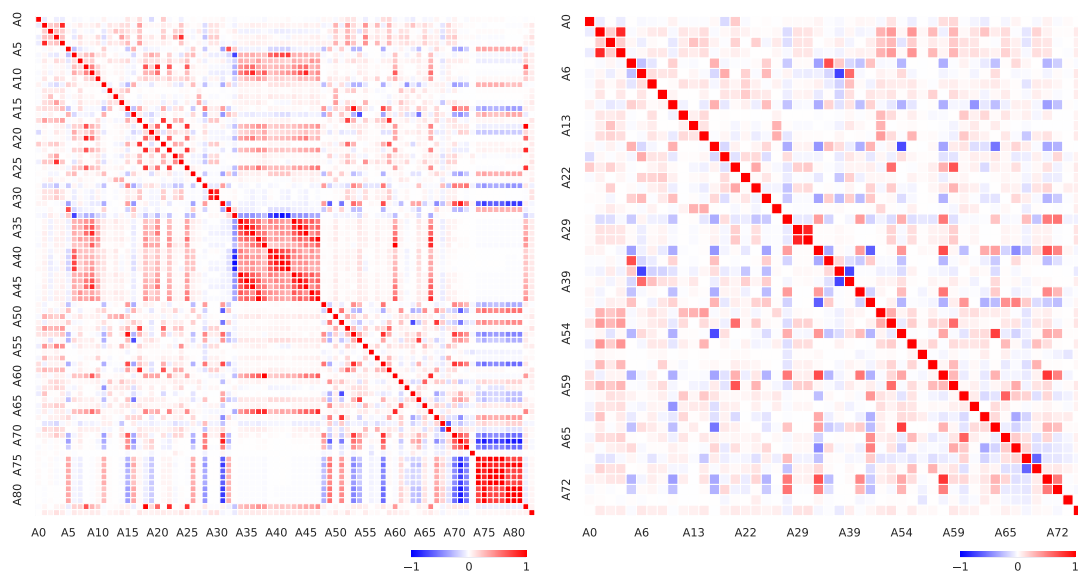


图 3-6 因子关联系数矩阵。左侧为所有因子的关联系数矩阵，右侧为经过筛选后因子的关联系数矩阵。

本文共构造了 83 个因子，标记为 $(A_0, A_1, \dots, A_{82})$ 。表 3-2 展示了前十个因子的构造方法，使用的数据类型，以及计算的时间复杂度。图 3-5 展示了 83 个因子与 1 分钟收益率的分布直方图。

得到因子数据后，需要对因子进行筛选。传统的方式是先计算因子间的关联系数，然后根据选定的阈值，仅保留关联性高于这一阈值的因子对中的一个因子。也可以直接对每个因子单独进行回归，得到单个因子对目标收益率的解释程度，保留解释性较好的因子。为了进一步降低运算成本，还可以对得到的因子进行多元回归，然后剔除解释性较弱的因子。

本文直接使用机器学习方法（见 3.3.3），对 83 个因子进行回归检测，根据重要程度，选取了其中 47 个因子。图 3-6 左侧展示了 83 个因子与 1 分钟收益率间的关联系数 (RankIC)，右侧展示了经过筛选后，得到的 47 个因子间的关联系数。经过筛选，有效的缓解了因子间的共线性问题，降低了算法的运算成本。

3.3.3 模型构建

本文使用 XGBoost 预测一分钟的股价收益率。XGBoost^[149]是一个基于 Boosting 的集成学习方法，采用决策树作为基学习器。相较原始的 Gradient Boosting，XGBoost 优化了寻找决策树中分裂节点的方法，降低了过拟合风险，提高了标准 Gradient Boosting 算法的训练效果。

3.3.3.1 XGBoost 原理

对于包含 n 个样本 m 个特征的数据集 $\mathcal{D} = \{(x_i, y_i)\} (|\mathcal{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$, 由 K 个决策树组成的集成模型的输出可以表示为,

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i) \quad (3-5)$$

其中, $f_k(x_i)$ 为第 k 个基学习器的输出。XGBoost 的目标函数为,

$$\arg \min \mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (3-6)$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2. \quad (3-7)$$

其中, l 为损失函数, Ω 是一个附加的正则项, 用来平滑最终学到的权重, 以防止过拟合。 Ω 为 0 时, 模型退化为原始 Gradient Boosting。正则项中的 T 为决策树中叶节点的数量, w 为叶节点的权重, γ 和 λ 分别是控制叶节点数量和权重的惩罚项, $\|w\|^2$ 表示权重的 L2 距离。

在训练时, 经过 t 次迭代, 样本 i 的预测值可以写为,

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \quad (3-8)$$

其中, f_t 是第 t 次迭代的新建树, η 是学习率, 也称为收缩尺度。使用较小的学习率可以降低每棵决策树对最终预测的影响, 减少过拟合风险并提高模型的泛化效果, 但需要使用更多的决策树才能实现良好的性能表现。增大学习率则会提高每棵决策树对最终预测结果的影响, 有助于减少决策树的使用数量, 加快训练的过程, 但也会增加过拟合的风险。为简化讨论, 下文令 $\eta = 1$ 。此时的训练的目标为最小化如下目标函数,

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t). \quad (3-9)$$

上式对 $\hat{y}_i^{(t-1)}$ 展开至二阶有,

$$\mathcal{L}^{(t)} = \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t). \quad (3-10)$$

其中,

$$g_i = -\frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \quad (3-11)$$

用 I_j 表示叶节点 j 所包含的样本集, 消去常数项并带入公式3-7后, 得到约化的目标函数,

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \\ &= \sum_{j=1}^T \left[G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T. \end{aligned} \quad (3-12)$$

其中, $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$ 。根据上式, 可以计算叶节点 j 的最优权重, 以及对应的目标函数,

$$w_j^* = -G/(H + \lambda), \quad \tilde{\mathcal{L}}^{(t)} = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T. \quad (3-13)$$

XGBoost 通过贪心算法寻找决策树的最佳分割。计算某个分割的如下收益,

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (3-14)$$

其中, 下标 L , R 分别代表左、右子节点的对应值, 在收益最大时进行分割。实际训练时, Chen et al. [149]提供了精确与近似两种算法处理分割问题, 并且考虑了稀疏样本时的情况。

3.3.3.2 训练过程

本文使用的数据集为 2023 年 2 月的 A 股全市场交易数据。在处理数据时, 首先将其分为了两个部份: 80% 作为训练集, 用于模型的训练和调优; 剩余的 20% 则作为测试集, 用于评估模型的预测性能。模型的训练一共分为三个阶段:

- 第一阶段是因子筛选。在这一阶段, 本文对数据集中所有 83 个因子进行了分析, 以确定哪些因子对于预测股价有显著的影响, 目的是减少模型的复杂

度并提高其预测的准确性。这一步是通过对所有因子进行预训练实现的。根据训练的结果，最终选取了 47 个因子。

- 第二阶段是模型参数的调试。针对选定的因子构建预测模型后，需要对其参数进行细致的调整，包括学习率的选择、树的深度、正则化参数的设定、以及其他可能影响模型性能的参数等。通过对这些参数的调整，确保模型在训练集上达到较好的拟合效果，同时避免过拟合。
- 第三阶段是最终预测模型的获得。在经过前两个阶段的准备之后，使用调整好的模型对训练集数据进行了最后一次拟合。这一次拟合的目的是综合前两阶段的工作，得到一个既能够很好地反映数据特性，又具有较高预测准确度的模型。

3.3.4 训练结果及分析

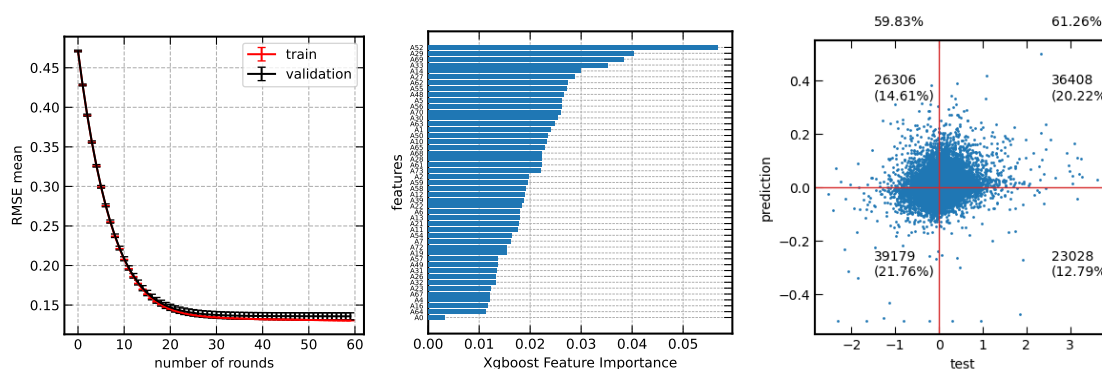


图 3-7 左侧展示了训练过程的收敛性。中间展示了特征对模型贡献的重要程度。右侧展示了模型在测试集上真实值与预测值的分布。

图3-7展示了模型的收敛性（左），以及各因子的重要程度（中）。从图中可以看出，模型在测试集和验证集上表现出了良好的收敛性能，表明模型能够有效地预测未知数据并且避免了过拟合风险。

训练结束后，我们使用测试集对模型性能进行了评估。图3-7右侧展示了模型预测的准确度，横坐标为测试集的真实值，纵坐标为对应的预测值。仅以价格变化方向作为基准，对于涨跌预测的准确度 (accuracy) 分别为 61.26% 和 59.83%，整体的预测准确度为 60.51%。

在评估模型表现时通常需要关注以下几个方面：（1）买卖信号的分布：信号的密集程度可以反映出算法对市场波动性的反应。信号过于密集可能意味着策略对市场噪音的过度敏感，而信号稀疏则可能意味着策略较为保守。此外，买卖信号是否出现在价格波动的极点，也是判断算法效能的一个重要指标。（2）价格趋势

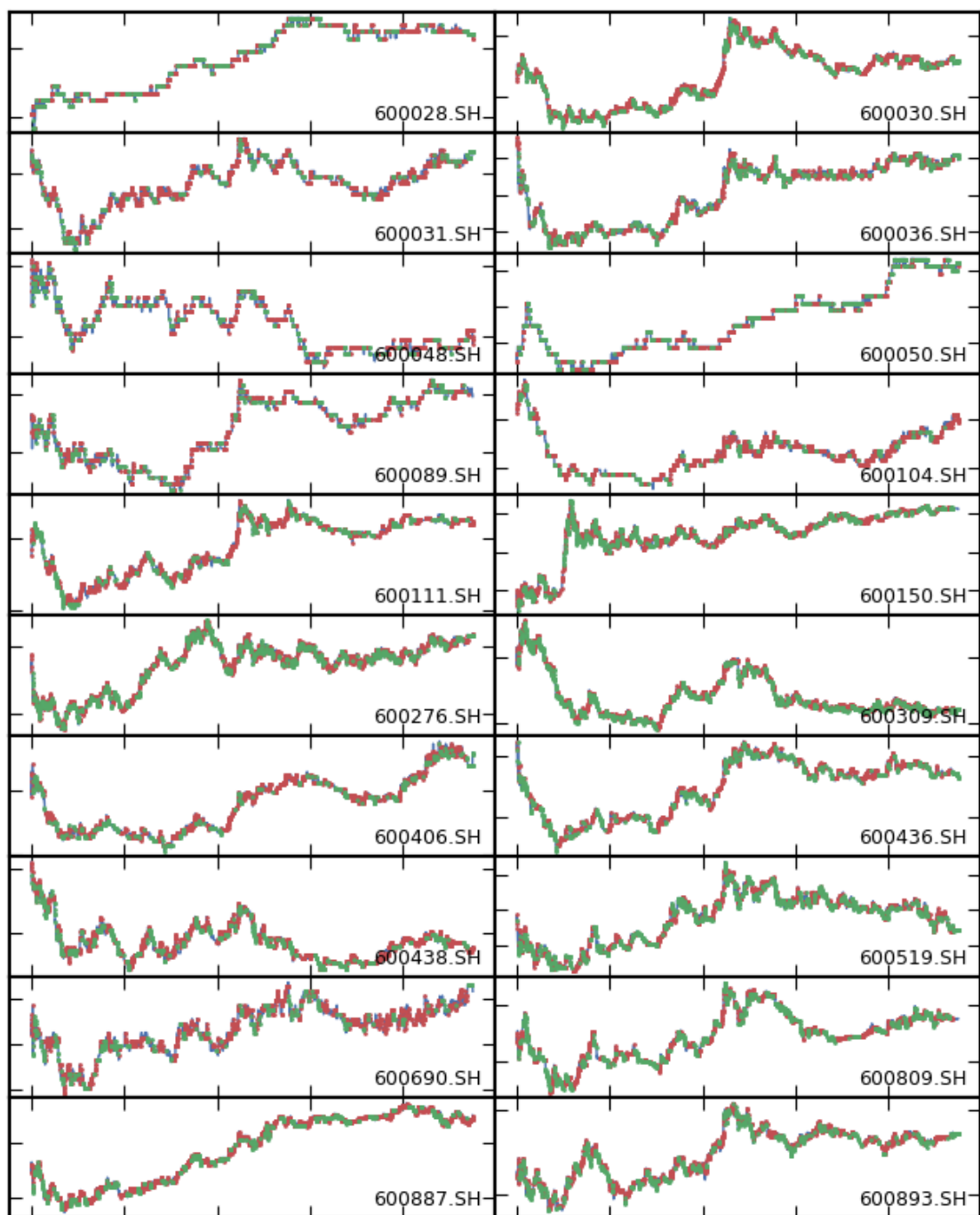


图 3-8 信号分布展示。选取了上证 50 指数的部份成分股。红色数据点为买入点，绿色数据点为卖出点。

与算法信号的一致性：理想情况下，买入信号应该出现在价格上升趋势之前，卖出信号应该出现在价格下降趋势之前。如果信号与趋势不一致，可能表明算法需要进一步优化。（3）不同股票之间的比较：通过比较不同股票的走势和算法信号，

可以分析算法是否具有泛化能力，即能否在不同的股票上都实现有效的交易。(4) 交易频率与市场波动的关系：观察不同市场波动条件下的交易频率，可以帮助我们了解算法是否能够适应市场环境的变化。(5) 算法的适用性分析：结合具体的市场环境和股票特性，分析算法是否适用于不同的市场或股票类型，例如，周期性行业股票与成长性行业股票可能会有不同的表现。

图3-8展示了上证 50 成份成分股在某一交易日内的价格走势，以及由交易策略产生的买入和卖出信号。每个子图代表一个不同的股票，图中的代码如 600028.SH、600030.SH 等，代表着上海证券交易所 (.SH) 不同上市公司的股票代码。每个子图中，红色和绿色的数据点分别标记了算法策略认定的买入（红色）和卖出（绿色）的所有可能时机。

通过调整交易信号的阈值，可以很容易的调节信号的密集程度；从价格趋势来看，买入信号大多出现在价格上升区间，表明信号与价格趋势呈现正相关；对于上证 50 成份股，信号表现基本一致，模型具有一定的泛用性。

3.4 结论与讨论

本章比较全面的介绍了一种基于机器学习的高频算法交易策略，研究探讨了机器学习在开发实时自适应算法交易策略中的整合应用。

本章从金融市场的演变和高频算法交易的兴起开始，详细描述了本文使用的交易系统结构，包括系统内的各个组件和其相互作用。随后基于国内 A 股市场粒度级别数据，详细介绍了一种算法交易策略的研究方法。包括交易策略的定义、数据处理技术（包括噪声过滤和因子计算）以及特征工程，并重点讨论了使用历史市场数据构建和测试交易策略的过程。

具体来说，本文主要使用 XGBoost 集成学习方法，介绍了一个针对国内 A 股的算法交易策略的构建过程。首先，概述了 XGBoost 的原理，之后，深入模型构建过程的细节，包括预训练，参数调试，减少过拟合风险等提高训练效果的方式。最后展示了模型在测试集上的结果和分析，包括模型的收敛性、不同因子的重要性、以及模型的预测的准确性。结果分析还包括了买卖信号随价格趋势的变化，以及算法在不同股票下的一致性和适用性的考察。

后续，我们将针对以下几点策略做进一步的优化和更新：首先是数据处理和噪声过滤的具体方法可能需要进一步细化。其次，需要对模型在不同市场条件下的适应性进行深入分析。另外，应当进一步探索不同机器学习算法的表现，并于与当前方法进行比较，不断优化算法。最后，本文略去了在实际交易中模型的表现，包括应用潜力和限制等，这些将是后续分析的重点。

第四章 基于强化学习的中国 A 股 T+0 策略研究

T+0 策略是指对某个投资标的进行当日买入，当日卖出的操作。对冲基金或投资经理通过使用 T+0 策略提高资金的使用效率和流动性，使得投资者可以利用市场的短期波动获取更多收益。本章主要介绍作者开发的基于 A 股股票市场的 T+0 交易策略。

4.1 研究背景

在北美、香港及期货等市场中，大多数投资品类都实行 T+0 交易制度，即允许投资者在同一交易日内完成同一支股票的买入和卖出。这种制度为市场参与者提供了较高的灵活性和即时性，有利于提高市场流动性和效率。相比之下，中国 A 股市场自 1995 年起开始实行 T+1 交易策略。即投资者在购买某支股票的当天不能卖出该支股票，只能在第二个交易日进行卖出操作。这一政策的初衷在于保证股票市场的稳定，防止过度投机行为，从而在一定程度上维护市场秩序和投资者利益。

为了在 A 股市场实现类似 T+0 交易的效果，目前普遍可以采用两种策略。第一种方法是持有底仓策略，即投资者在交易日之前已持有某股票头寸，然后在交易日卖出前一交易日的持仓，从而实现 T+0 交易的效果。这种方法操作简单，但存在一定风险：由于需要承担隔夜市场波动的影响，可能会面临不可预测的市场变化。第二种方法是通过融资融券实现 T+0 交易。融资融券，即投资者通过向证券公司借入资金或证券进行交易。这种方式可以绕过 T+1 交易机制的限制，实现当日买卖¹。我国自 2006 年开始实施融资融券制度。初始阶段，只有 10 只股票可以进行融资融券交易。经过多年的发展和市场调整，截至 2023 年，融资融券的标的股票数量已经扩大至 1800 多只。这一变化显著提高了市场的灵活性和投资者的选择范围。相较于持有底仓策略，融资融券可以有效降低市场的隔夜风险，但同时也带来了额外的借贷成本，并且可交易的标的受到限制。

总的来说，虽然中国 A 股市场实行 T+1 交易策略，但通过持有底仓和融资融券等方式，投资者仍然能够在一定程度上实现类似 T+0 交易的效果。在此背景下，结合现有的算法交易策略，本章的主要目的是开发一种适用于中国 A 股市场的 T+0 交易策略，其核心目的是利用先进的算法和技术手段优化现有的交易机制。通过允许投资者在同一交易日内完成买入和卖出操作，提高交易的灵活性和效率，同

¹ 上海证券交易所融资融券交易实施细则

时有效降低算法交易的风险并提高投资回报。该策略采用强化学习作为主要的技术手段，运用大量的股票分钟级数据来预测股票价格趋势，从而更准确地捕捉市场动态。

本章首先将详细介绍股票分钟数据的处理方法，包括数据清洗、规范化及特征工程，确保数据质量和有效性。接着，将进一步阐述所选用模型的定义和构建方法，包括强化学习模型的框架、算法逻辑及其在股票交易中的应用方式。之后，将详细介绍模型的训练过程，以及模型预测的准确性和可靠性。最后，为了全面评估所提出策略的实际应用价值，本文将展示模型的回测结果。

4.2 研究方法

4.2.1 数据处理

本文假定 T+0 的交易时段为全天自由竞价时段。为了预测更长周期内的价格走势，并降低运算成本，本文使用分钟数据作为数据源。

4.2.1.1 分钟数据合成与收益计算

数据处理的的第一步是利用逐笔数据合成分钟数据，包含每分钟的开盘、最高、最低、以及收盘（OHLC）价格，以及每分钟内的交易量。图4-1是利用上证 50 逐笔数据合成后的分钟数据，为了方便展示，图中仅显示了每分钟的收盘价格（蓝色实线）以及每分钟内的交易量（柱状图）。本文研究的数据集主要为上证 50 指数成份股 2023 年 1 到 6 月的分钟数据。

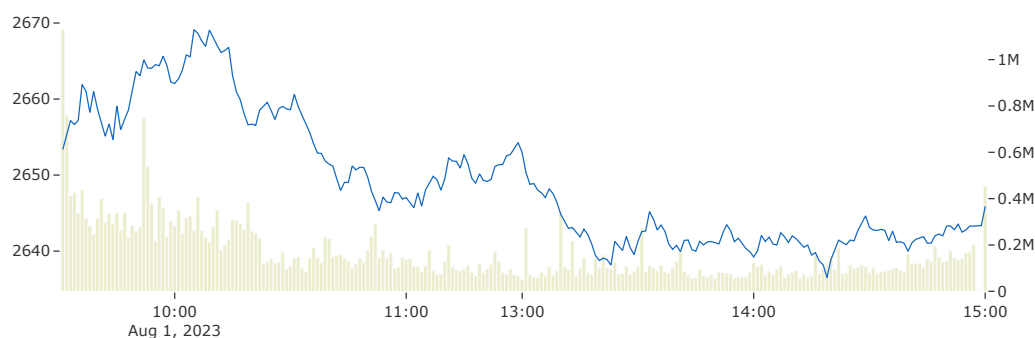


图 4-1 上证 50 指数一天的分钟级数据。为方便展示，图中仅显示了每分钟的收盘价格（实线）。柱状图为对应 1 分钟内的交易量。

本文使用如下公式计算证券收益率，

$$r(t, \Delta t) = \log P(t + \Delta t) / P(t) \quad (4-1)$$

其中， $P(t)$ 为 t 时证券价格， $P(t + \Delta t)$ 为 Δt 时的价格， $r(t, \Delta t)$ 为证券在 t 时刻在经过 Δt 后对应的收益率。我们分别计算了数据集中每个点对应的 2 到 5 分钟后的收益率，即 $\Delta t = 2, 3, 4, 5 \text{ min}$ 。得到的收益率分布如图 4-2。左侧为不同 Δt 收益率分布比较，从上到下 Δt 逐渐增大，不同颜色的深度表示分布的密度大小，颜色越深，密度越大。右侧单独展示了 5 分钟收益率的分布曲线。从收益率的分布图可以看出，收益率分布不是高斯分布，且存在偏置^[207-209]。本文的预测对象为 5 分钟收益率走势。

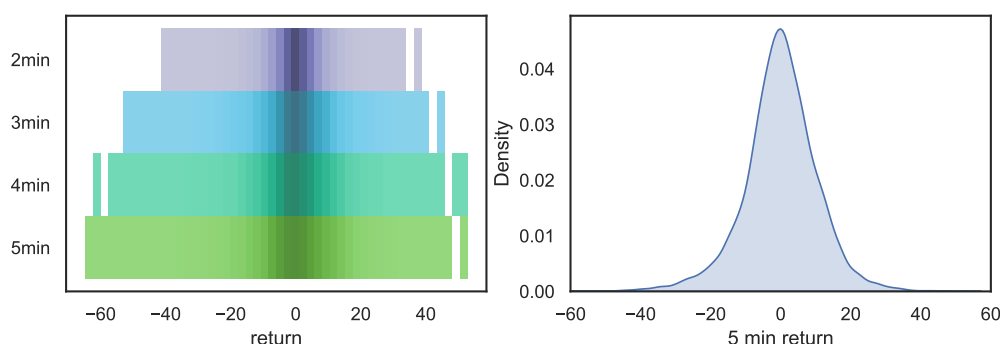


图 4-2 上证 50 指数收益率分布。左图为 2、3、4、5 分钟收益率分布的比较，右图为 5 分钟收益率分布曲线。

4.2.1.2 特征工程

本文基于分钟级股票交易数据，实现了 42 个技术指标，用于描述股票的交易状态。这些指标涵盖了从价格动量到交易量变化等多个维度。具体的指标列表及其缩略名和详细描述参考表格 4-1。

本部份使用上证 50 指数作为具体的研究对象，用以说明特征工程的处理过程。首先对这 42 个技术因子进行了严格的数据清洗和正则化处理，以确保数据的准确性和一致性。图 4-3 的左侧部分展示了这些因子在处理后的数值分布情况。接下来，为了探究这些技术因子与股票短期收益之间的关系，我们计算了它们与 5 分钟收益率之间的滚动相关性。此处，我们选择 5 分钟作为滚动窗口，以捕捉更为精细的市场动态。图 4-3 右侧部份展示了这些滚动关联函数的分布情况。基于这一结果，我们优先选取了那些与股价收益率相关性高且一致性强（关联性系数中位

short	name		short	name	
0	ADL	Accumulation Distribution Line	14	KO	Klinger Oscillator
1	ASI	Accumulative Swing Index	15	MFI	Money Flow Index
2	ABEWI	Ar Br Emotion Willingness Index	16	MACD	Moving Average Convergence Divergence
3	ARO	Aroon Oscillator	17	NVI	Negative Volume Index
4	ATR	Average True Range	18	PSAR	Parabolic SAR
5	BBW	Bollinger Band Width	19	%R	percent R
6	CO	Chaikin Oscillator	20	PVI	Positive Volume Index
7	CMF	Chaikin Money Flow	21	PVT	Price Volume Trend
8	CMO	Chande Momentum Oscillator	22	QST	Qstick
9	CCI	Commodity Channel Index	23	ROC	Rate Of Change
10	ADX	Average Directional Movement Index	24	ROCHL	Ratio OCHL
11	EOM	Ease Of Movement	25	RSI	Relative Strength Index
12	DTUV	Down To Up Volatility	26	RVI	Relative Volatility Index
13	IMI	Intraday Momentum Index	27	SMI	Stochastic Momentum Index
28	SO	Stochastic Oscillator	34	VAR	Value At Risk
29	TC	Trend Count	35	VHF	Vertical Horizontal Filter
30	TEMA	Triple Exponential Moving Average	36	VAMA	Volume Adjusted Moving Average
31	TSI	True Strength Index	37	VMACD	Volume Moving Average Convergence Divergence
32	UI	Ulcer Index	38	VO	Volume Oscillator
33	UO	Ultimate Oscillator	39	VR	Volume Ratio
40	VRSI	Volume Relative Strength Index	41	VIDA	Variable Index Dynamic Average

表 4-1 状态因子列表。

数 > 0.5) 的因子。

为了降低模型中因子间的共线性问题，同时提高模型计算效率，我们进一步对这些技术因子进行了筛选。图4-4左侧展示了各因子之间的相关性矩阵。我们剔除了那些与其他因子关联度较高的因子，最终确定了 10 个关键技术因子用于进一步的模型构建。图4-4右侧展示了最终 10 个因子间的相关性矩阵。

4.2.2 模型构建

在强化学习模型中，对股价走势预测的准确性进行优化，涉及到一系列关键变量的定义，包含状态空间、行动空间、和奖励函数等。这些变量是模型理解和分析股市动态的基础，本节将详细介绍这些变量的定义和处理的步骤。

4.2.2.1 状态空间

强化学习中的状态空间 S 理论上需要包含所有智能体可能遇到的情况，在股票交易中，可以由股票的历史价格、各种技术指标（因子）等进行描述。我们使用股票自身分钟线 OHLC 数据、交易量数据、以及经过特征工程得到的 10 个技术因子，作为描述股票状态的状态空间。

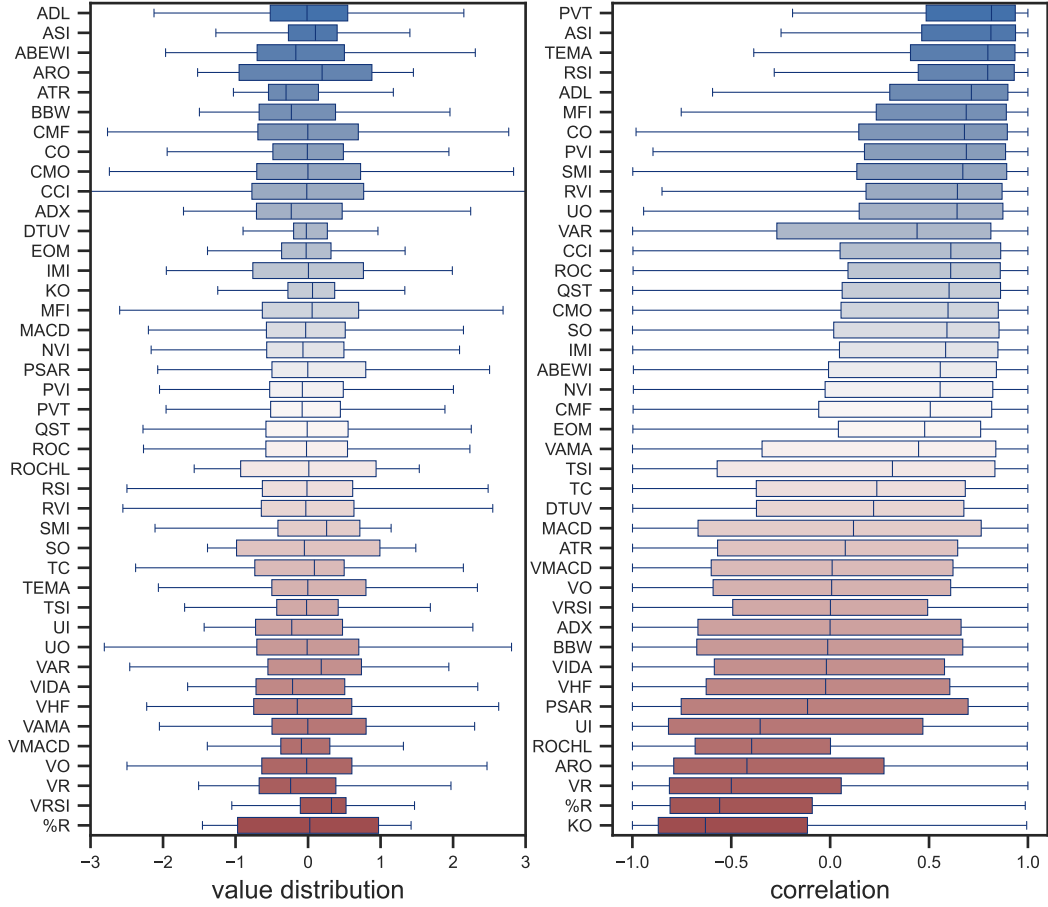


图 4-3 分钟因子分布（左）及其与 5 分钟收益率的 Spearman 滚动关联系数分布（右）。

4.2.2.2 行动空间

行动空间 A 是指智能体在每一步可以做的事情的集合。对于股票交易而言，行动空间包含买入、卖出或持有股票这三种动作。本文使用离散集合 $A \equiv \{1, 0, -1\}$ 描述买入、持有、以及卖出这三个动作。对任一行动 a ，有 $a \in A$ 。

4.2.2.3 奖励函数

奖励函数用来评估智能体所采取的行动。在股票交易中，奖励通常基于交易的盈利或亏损来定义。如果一个动作导致利润增加，则给予正奖励；如果导致亏损，则给予负奖励。假设每次的交易成本为 c ， t 时刻采取行动 a_t 的奖励函数可以表示为，

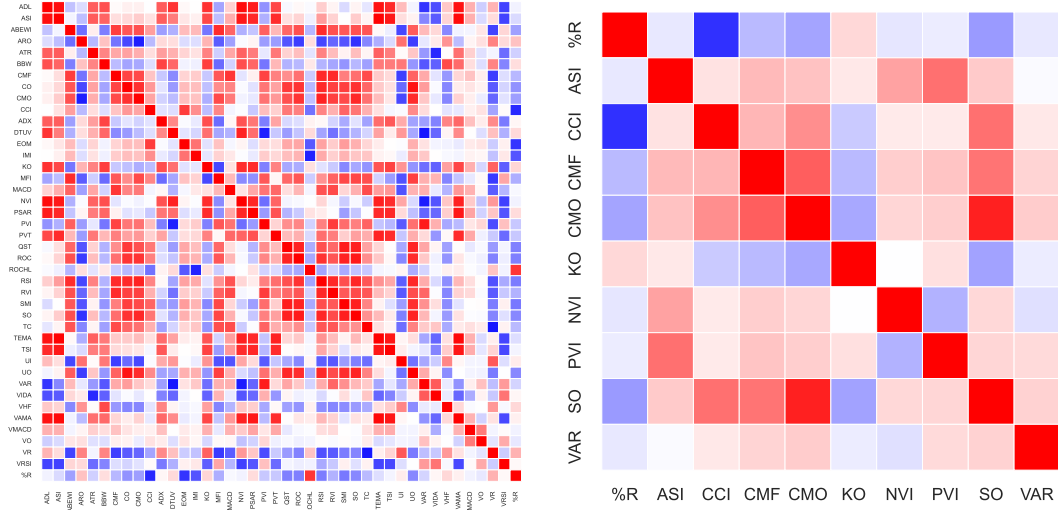


图 4-4 分钟因子 Spearman 关联系数矩阵。

$$R(a_t, t) = \begin{cases} P(t+1) - P(t) - c & \text{if } a_t = 1 \text{ (买入)} \\ P(t) - P(t+1) - c & \text{if } a_t = -1 \text{ (卖出)} \\ 0 & \text{if } a_t = 0 \text{ (持有)} \end{cases} \quad (4-2)$$

在这个公式中：当 $a_t = 1$ （买入）时，奖励是下一时刻的价格 $P(t+1)$ 减去当前时刻的价格 $P(t)$ ，再减去交易成本 c ；当 $a_t = -1$ （卖出）时，奖励是当前时刻的价格 $P(t)$ 减去下一时刻的价格 $P(t+1)$ ，再减去交易成本 c ；当 $a_t = 0$ （持有）时，奖励为 0，表示在该时刻没有进行交易。需要注意的是，为了简化讨论，上述奖励函数中假设了交易成本是固定的，而在实际情况中，交易成本与交易量、交易类型或市场条件等因素相关。

4.2.2.4 深度 Q- Learning 模型构建

在第二章中我们介绍了强化学习的相关概念，包含值函数与策略函数等，以及常见的强化学习方法，如蒙特卡洛、时序差分学习等。其中，深度强化学习（DQN）是利用深度神经网络来逼近值函数和策略函数的一类学习方法。本文将使用 DQN 构建强化学习模型。

DQN 模型包含以下几个重要部份：

- Q 网络结构。DQN 使用深度神经网络来近似 Q 函数，即状态-动作对的价值

函数。这里我们使用简单的多层全连接神经网络，用于估计在给定状态下执行每个动作的预期回报。网络的输入是状态空间的维度，输出是动作空间的维度。

- 经验回放 (Experience Replay)。为了打破样本之间的相关性并避免对最近的经验过拟合，DQN 使用经验回放方法。具体来说，DQN 将经验（状态、行动、奖励、新状态）存储在一个回放缓冲区中，然后从中随机抽取小批量样本进行训练。本课题中历史经验为交易者的状态、买卖行为、以及累积收益。
- 固定 Q 目标 (Fixed Q-targets)。DQN 通过使用两个结构相同但参数不同的神经网络来解决训练过程中的目标变化问题。一个是用于预测 Q 值的网络（在线网络），另一个用于计算目标 Q 值的网络（目标网络）。目标网络的参数定期从在线网络更新。

4.2.2.5 训练过程

DQN 的算法流程如下。

Algorithm 1 DQN 算法流程^[138]

```

初始化一个容量为  $N$  的经验回放记忆池  $\mathcal{D}$ 
随机初始化  $Q$  网络
for episode = 1,  $M$  do
    初始化序列  $s_1 = \{x_1\}$  和经过预处理的序列  $\phi_1 = \phi(s_1)$ 
    for  $t = 1, T$  do
        以概率  $\epsilon$  选择一个随机行动  $a_t$ （探索模式）
        否则选择令  $Q$  最大的行动,  $a_t = \max_a Q^*(\phi(s_t), a; \theta)$ （经验模式）
        在仿真器中执行动作  $a_t$  观察获得的奖励  $r_t$  以及  $x_{t+1}$ 
        令  $s_{t+1} = s_t, a_t, x_{t+1}$  并预处理  $\phi_{t+1} = \phi(s_{t+1})$ 
        存储  $(\phi_t, a_t, r_t, \phi_{t+1})$  到经验池  $\mathcal{D}$ 
        从经验池  $\mathcal{D}$  中进行随机小批量采样  $(\phi_j, a_j, r_j, \phi_{j+1})$ 
        令  $y_j = \begin{cases} r_j & \text{对于终止状态 } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{对于非终止状态 } \phi_{j+1} \end{cases}$ 
        对  $(y_j - Q(\phi_j, a_j; \theta))^2$  执行梯度下降步骤，更新  $Q$  网络
    end for
end for
    
```

本文使用 PyTorch 库实现上述 DQN 算法，利用上证 50 指数成份股 2023 年 1 到 5 月的数据作为训练集，然后在 6 月的数据上对模型的预测进行了评估。

4.2.2.6 交易信号生成

高频数据能够提供更细粒度的信息，例如短期波动，而低频数据则可能更好地反映长期趋势和周期性。为了提高预测的准确性和鲁棒性，我们将 5 分钟的预测

信号与上一章中算法交易策略的信号进行了结合，得到最终的交易信号。

4.2.3 T+0 选股

并非所有股票都适合进行 T+0 交易，一般来说，具有以下特征的股票更适合进行 T+0 交易：

- 充足的流动性：流动性高的股票意味着买卖订单可以迅速成交，减少了进入和退出头寸的时间成本。高流动性通常与较大的交易量和较小的买卖价差相关。
- 中高价格波动性：适度的价格波动性可以为 T+0 交易提供充足的利润空间。但是，过高的波动性，例如波动性激增时，可能会增加交易风险。
- 明显的市场趋势：处于明显上升或下降趋势中的股票提供了更清晰的交易方向，更适合于短期交易。此外，大多数交易者会寻找与其行业或指数相关联的股票，通常当指数或行业上涨（下跌）时，与其对应的个股价格也会上涨（下跌）。
- 良好的公司基本面：具有稳健财务状况和良好增长前景的公司，其股票价格更可能反映真实价值，通常是更好的 T+0 交易选择。

根据上述特点，本文使用自组织映射（self-organizing map, SOM）算法，对股票进行分类，并从中筛选出 T+0 交易效果更好的股票，生成 T0 股票池。

SOM 是由芬兰教授 Teuvo Kohonen 提出的一种人工神经网络^[210]，它通过自学习过程来组织信息，使得类似的数据在网络中彼此靠近。SOM 包含一个由神经元组成的网格，每个神经元都与输入数据中的特征相连接。在训练过程中，输入数据呈现给网络，并找到与输入最相似的神经元（胜者神经元）。然后调整胜者神经元及其邻域内神经元的权重，使其更接近输入数据。随着重复的训练迭代，网络学会识别数据中的模式和聚类。

具体来说，假设输入的状态因子为， $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ ，且第 i 个因子与第 j 个神经元的权重为 w_{ij} 。定义判别函数，

$$D(j) = \sum_i^n (w_{ij} - x_i) \quad (4-3)$$

寻找使得 $D(j)$ 最小的神经元 $\hat{j}$ ，即为胜者神经元。对每个与胜者神经元 $\hat{j}$ 临近的其他神经元，以及所有的 x_i ，更新权重。重复上述步骤，将输入进行聚类。然后应用于前面的 T+0 策略，找出表现最好的一组作为 T0 股票池。

图4-5展示了利用 SOM，根据股票特征对其进行分类的结果。训练集为 A 股所

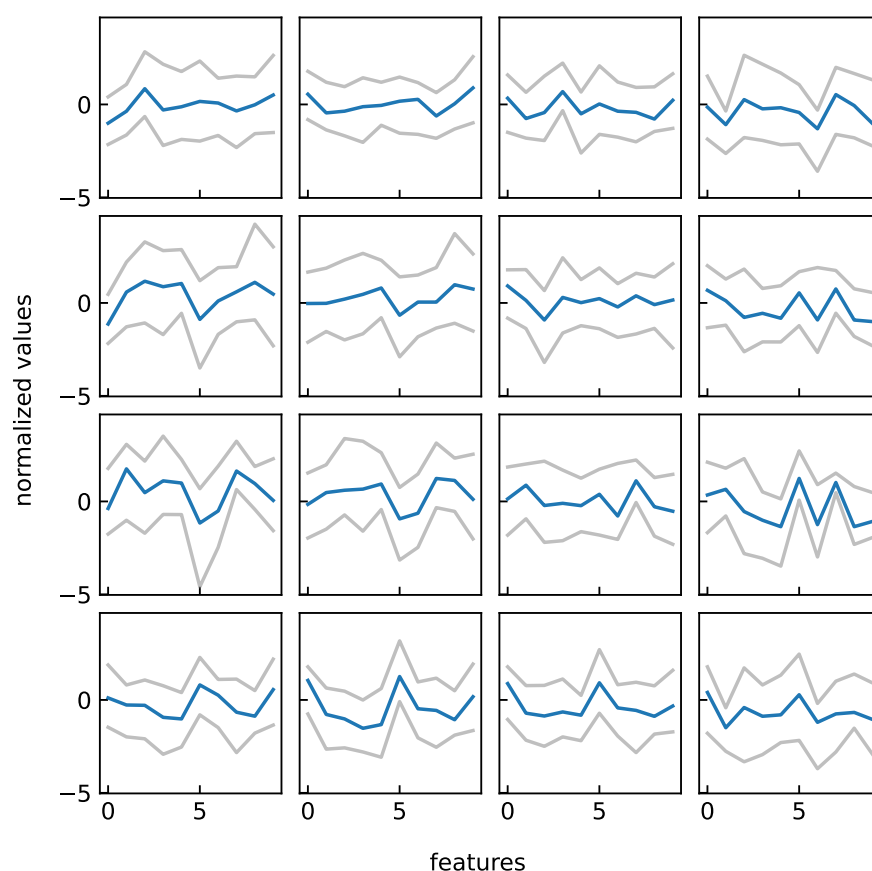


图 4-5 利用 SOM，根据股票特征对其进行分类的结果。一共展示了 16 种结果。

有股票 2023 年第一季日频数据。这里将股票根据所选特征的特点一共分为了 16 组，蓝色实线为特征的平均值，灰色实线对应特征在该类型下的最大与最小值。

4.3 交易结果及分析

对于 T+0 交易，止盈止损对于风险管控至关重要。常见的设置止盈止损点的方法有：

- (1) 根据预设的回报风险比设置止盈止损点；
- (2) 根据历史价格波动设置止盈止损点；
- (3) 通过其他技术分析设置止盈止损点。

本文采用了简单的固定点数的方式设置了止盈止损点，在入场价下跌 2% 时触发止损，在入场价上升 2% 时触发止盈。此外，交易策略必须符合所有监管要求，例如融资融券时卖单不可以低于市价订单、每笔交易金额的上限、每天交易笔数的上限等。

设定好止盈止损点以及监管条件限制后，我们使用上证 50 成分股对策略进行了模拟盘测试。图4-6展示了利用 2023 年 6 月数据进行测试的统计结果。横坐标对应不同类型的股票，纵坐标为该类股票的收益分布。从图中可以看出，第 3 类与第 6 类股票在该策略下，具有可观切稳定的收益表现。

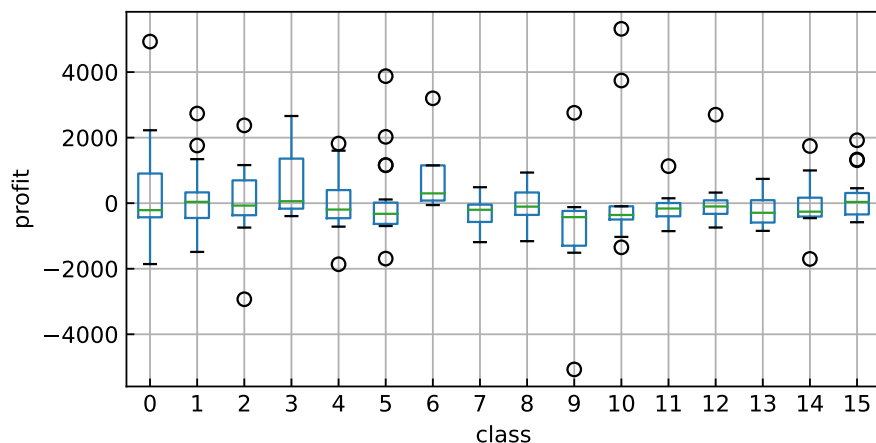


图 4-6 上证 50 成分股 2023 年 6 月各分类股票的 T+0 收益分布。

4.4 结论与讨论

本章介绍了一种基于强化学习的中国 A 股 T+0 交易策略，展示了利用先进的机器学习技术优化交易过程和提高投资回报的可能性。具体来说，本章基于股票的分钟级数据，详细介绍了特征工程、模型构建和训练、交易策略的实施、以及 T+0 选股等过程，提出了一种能够适应中国特有的 T+1 交易制度，并在一定程度上模拟 T+0 交易效果的策略。

该策略通过使用 42 个技术指标作为特征工程的基础，结合强化学习模型，研究预测了股票价格的短期走势。证明了在处理高频交易数据时，综合利用多种技术指标及深度学习技术的有效性。此外，策略还利用自组织映射（SOM）算法对股票进行了分类，能够有效地筛选出最适合进行 T+0 交易的股票。这一过程不仅提高了交易的成功率，也为投资者提供了一个科学且实用的选股工具。在止盈止损点的设定上，策略采用固定收益点数的方法，在保证策略的可操作性和稳定性的基础上，简化了风险控制过程。通过对上证 50 成分股的模拟盘测试，证明了该策略在实际市场中具备一定的有效性。

尽管该策略取得了一些初步成果，但在实际应用中仍面临一些挑战和限制。首先，市场效率和动态性是影响 T+0 交易策略的重要因素，例如，市场情绪、宏观

经济政策、行业动态等都可能影响策略的表现。因此，需要进一步地寻找调整和优化该策略的方法，以适应新的市场变化。其次，在实际交易中，交易成本和滑点会对策略的净收益产生显著影响。虽然我们在模拟盘中考虑了其中的部份因素，但在现实交易中，这些因素的影响可能更加复杂。未来的研究可以进一步探讨如何减少这些成本的影响。此外，A 股市场的监管政策的不断变化对 T+0 交易策略的实施具有直接影响。这要求我们密切关注政策动态，及时调整交易策略以符合监管需求。最后，强化学习和其他机器学习技术正在快速发展，本文所研究的方法和结果应当被视为一个动态的过程，后续需要根据技术的累积不断地进行更新和改进。

综上，本文研究的基于强化学习的中国 A 股 T+0 交易策略，展现出了良好的市场应用前景。随着市场环境的变化和先进技术的发展，我们会进一步对研究方法进行优化，以期未来能够提出更加有效、稳健的交易策略，为投资者创造更多价值。

全文总结

量化投资作为一种现代投资方法，通过使用数学模型、统计分析以及计算机技术来识别投资机会，从而最大化投资回报并控制风险。本文首先介绍了量化投资的两大核心组成部分：投资组合管理和算法交易。投资组合管理侧重于如何通过多元化投资来分散风险，以及如何利用量化模型来优化资产配置。算法交易则关注于如何使用算法来自动化交易决策过程，包括订单执行、市场监测等方面，以提高交易效率和执行质量。

随着人工智能技术的快速发展，机器学习算法在量化投资中的应用越来越广泛。本文第二部分详细介绍了几种常见的机器学习算法，包括传统机器学习算法（如线性回归、决策树）、深度学习、强化学习以及集成学习（如 XGBoost）。这些算法各有优势，能够处理不同类型的数据和投资问题。深度学习因其强大的数据处理能力，在处理复杂市场数据方面显示出巨大潜力；强化学习则以其在决策过程中的自我学习能力受到关注；集成学习通过组合多个模型来提高预测准确性，是提升算法交易性能的有效工具。

本文第三部份介绍了作者开发的一种基于 XGBoost 机器学习的高频算法交易策略。该策略通过分析大量历史数据来预测市场短期内的价格变动，以此生成交易信号。策略的开发过程包括数据收集、特征工程、模型训练以及策略回测等关键步骤。作者详细阐述了如何通过优化模型参数和增强模型的泛化能力来提高策略的稳定性和盈利性。

最后，本文针对国内 A 股市场的特点，提出了一种 T+0 交易策略。该策略利用深度强化学习和分钟级别的数据来预测股价走势，结合算法交易技术生成精准的交易信号。此外，该部份还探讨了 T+0 选股方法，旨在筛选出具有高交易潜力的股票。作者展示了如何通过实时数据处理、模型训练和策略优化等步骤，来适应 A 股市场的动态变化，从而在确保风险可控的同时，寻求超额回报。

整体而言，通过深入分析和实践，本文不仅展现了量化投资的理论框架和应用方法，而且通过具体案例说明了如何将先进的机器学习技术应用于实际的量化交易策略中。量化投资领域的持续研究和技术创新，为投资者提供了更多的机会，同时也带来了新的挑战。未来，随着技术的进步和市场环境的变化，量化投资策略将继续演化，为投资管理实践带来更多的可能性。

附录 A 附表

Status	Meaning	Status	Meaning
null	正常交易	'O'(TRADE)	连续撮合
0	首日上市	'P'(BREAK)	休市
1	增发新股	'Q'(VOLA)	波动性中断
2	上网定价发行	'R'(BETW)	交易间
3	上网竞价发行	'S'(NOTRD)	非交易服务支持
'A'	交易节休市	'T'(FCALL)	固定价格集合竞价
'B'	整天停牌	'U'(POSTR)	盘后处理
'C'	全天收市	'V'(ENDTR)	结束交易
'D'	暂停交易	'W'(HALT)	暂停
'E'(START)	启动交易盘	'X'(SUSP)	停牌
'F'(PRETR)	盘前处理	'Y'(ADD)	新增产品
'G'(DEL)	不可恢复交易的熔断阶段（上交所的 N）	'Z'(DEL)	可删除的产品
'H'(HOLIDAY)	放假	'd'	集合竞价阶段结束到连续竞价阶段开始之前的时段（如有）
'T'(OCALL)	开市集合竞价	'q'	可恢复交易的熔断时段（上交所的 M）
'J'(ICALL)	盘中集合竞价	'v'	市场波动调节机制（港交所）
'K'(OPOBB)	开市订单簿平衡前期	'<'	盘前
'L'(IPOBB)	盘中订单簿平衡前期	'>'	盘后
'M'(OOBB)	开市订单簿平衡	'c' (小写 c)	收市竞价交易（Closing Auction Session (CAS)）
'N'(IOBB)	盘中订单簿平衡	'e'	有盘后交易的品种的正常交易时间结束（比如科创板 15:00 时交易结束）

表 A-1 状态说明

参考文献

- [1] Meucci A. 'P' Versus 'Q': Differences and Commonalities between the Two Areas of Quantitative Finance[J]. GARP Risk Professional, 2011: 47-50.
- [2] Gomber P, Arndt B, Lutat M, et al. High-Frequency Trading[J]. SSRN Electronic Journal, 2011. DOI: 10.2139/ssrn.1858626.
- [3] Hasbrouck J, Saar G. Low-latency trading[J/OL]. Journal of Financial Markets, 2013, 16(4): 646-679. <https://www.sciencedirect.com/science/article/pii/S1386418113000165>. DOI: <https://doi.org/10.1016/j.finmar.2013.05.003>.
- [4] Li S, Wang X, Ye M. Who provides liquidity, and when?[J/OL]. Journal of Financial Economics, 2021, 141(3): 968-980. <https://www.sciencedirect.com/science/article/pii/S0304405X21001501>. DOI: <https://doi.org/10.1016/j.jfineco.2021.04.020>.
- [5] Jones C M. What Do We Know About High-Frequency Trading?[J/OL]. SSRN Electronic Journal, 2013. <https://doi.org/10.2139/ssrn.2236201>. DOI: 10.2139/ssrn.2236201.
- [6] Chung K H, Lee A J. High-frequency Trading: Review of the Literature and Regulatory Initiatives around the World[J/OL]. Asia-Pacific Journal of Financial Studies, 2016, 45(1): 7-33. <https://doi.org/10.1111/ajfs.12120>. DOI: 10.1111/ajfs.12120.
- [7] Miller R S, Shorter G. High Frequency Trading: Overview of Recent Developments[R/OL]. Washington D.C.: University of North Texas Libraries, UNT Digital Library, 2016. <https://digital.library.unt.edu/ark:/67531/metadc847719/>.
- [8] Brogaard J, Garriott C. High-Frequency Trading Competition[J/OL]. Journal of Financial and Quantitative Analysis, 2018, 54(4): 1469-1497. <https://doi.org/10.1017/s0022109018001175>. DOI: 10.1017/s0022109018001175.
- [9] Comerton-Forde C, Malinova K, Park A. Regulating dark trading: Order flow segmentation and market quality[J/OL]. Journal of Financial Economics, 2018, 130(2): 347-366. <https://doi.org/10.1016/j.jfineco.2018.07.002>. DOI: 10.1016/j.jfineco.2018.07.002.

-
- [10] Markowitz H. Portfolio Selection[J/OL]. The Journal of Finance, 1952, 7(1): 77-91 [2023-09-18]. <http://www.jstor.org/stable/2975974>.
- [11] Sharpe W F. CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK*[J/OL]. The Journal of Finance, 1964, 19(3): 425-442. <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.1964.tb02865.x>. DOI: 10.1111/J.1540-6261.1964.TB02865.X.
- [12] Treynor J. Toward a Theory of Market Value of Risk Assets[J]. Unpublished Manuscript, a Revised Version Was Published in Korajczyk, Robert, A, Ed., 1999, Asset Pricing and Portfolio Performance: Models, Strategy and Performance Metrics, Risk Books, London, 15-22, 1962.
- [13] Lintner J. Distribution of Incomes of Corporations Among Dividends, Retained Earnings, and Taxes[J/OL]. The American Economic Review, 1956, 46(2): 97-113 [2023-09-19]. <http://www.jstor.org/stable/1910664>.
- [14] Mossin J. Equilibrium in a Capital Asset Market[J/OL]. Econometrica, 1966, 34(4): 768-783 [2023-09-19]. <http://www.jstor.org/stable/1910098>.
- [15] Ross S A. The arbitrage theory of capital asset pricing[J/OL]. Journal of Economic Theory, 1976, 13(3): 341-360. <https://www.sciencedirect.com/science/article/pii/0022053176900466>. DOI: [https://doi.org/10.1016/0022-0531\(76\)90046-6](https://doi.org/10.1016/0022-0531(76)90046-6).
- [16] FAMA E F, FRENCH K R. The Cross-Section of Expected Stock Returns[J/OL]. The Journal of Finance, 1992, 47(2): 427-465. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1992.tb04398.x>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1992.tb04398.x>. DOI: <https://doi.org/10.1111/j.1540-6261.1992.tb04398.x>.
- [17] Carhart M M. On Persistence in Mutual Fund Performance[J/OL]. The Journal of Finance, 1997, 52(1): 57-82. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1997.tb03808.x>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1997.tb03808.x>. DOI: <https://doi.org/10.1111/j.1540-6261.1997.tb03808.x>.
- [18] Fama E F, French K R. A five-factor asset pricing model[J/OL]. Journal of Financial Economics, 2015, 116(1): 1-22. <https://www.sciencedirect.com/science>

- /article/pii/S0304405X14002323. DOI: <https://doi.org/10.1016/j.jfineco.2014.10.010>.
- [19] Connor G. The Three Types of Factor Models: A Comparison of Their Explanatory Power[J/OL]. *Financial Analysts Journal*, 1995, 51(3): 42-46. <http://dx.doi.org/10.2469/faj.v51.n3.1904>. DOI: 10.2469/faj.v51.n3.1904.
- [20] Chen N F, Roll R, Ross S A. Economic Forces and the Stock Market[J/OL]. *The Journal of Business*, 1986, 59(3): 383-403 [2023-10-20]. <http://www.jstor.org/stable/2352710>.
- [21] Guo J, Wang S, Ni L M, et al. Quant 4.0: Engineering Quantitative Investment with Automated, Explainable and Knowledge-driven Artificial Intelligence [EB/OL]. arXiv. 2023. <https://arxiv.org/abs/2301.04020>.
- [22] Jiang Z, Xu D, Liang J. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem[EB/OL]. arXiv. 2017. <https://arxiv.org/abs/1706.10059>.
- [23] Snow D. Machine Learning in Asset Management—Part 1: Portfolio Construction—Trading Strategies[J/OL]. *The Journal of Financial Data Science*, 2019, 2(1): 10-23. <https://doi.org/10.3905/jfds.2019.1.021>. DOI: 10.3905/jfds.2019.1.021.
- [24] Snow D. Machine Learning in Asset Management—Part 2: Portfolio Construction—Weight Optimization[J/OL]. *The Journal of Financial Data Science*, 2020, 2(2): 17-24. <https://doi.org/10.3905/jfds.2020.1.029>. DOI: 10.3905/jfds.2020.1.029.
- [25] Conlon T, Cotter J, Kynigakis I. Machine Learning and Factor-Based Portfolio Optimization[J/OL]. *SSRN Electronic Journal*, 2021. <https://doi.org/10.2139/ssrn.3889459>. DOI: 10.2139/ssrn.3889459.
- [26] Sharpe W F. Mutual Fund Performance[J/OL]. *The Journal of Business*, 1966, 39(1): 119-138 [2023-10-05]. <http://www.jstor.org/stable/2351741>.
- [27] Sharpe W F. Adjusting for Risk in Portfolio Performance Measurement: 2 [EB/OL]. 1975. <https://www.pm-research.com/content/iijpormgmt/1/2/29>.
- [28] Modigliani F, Modigliani L. Risk-Adjusted Performance: 2[EB/OL]. 1997. <https://www.pm-research.com/content/iijpormgmt/23/2/45>.

- [29] Treynor J L, Black F. How to Use Security Analysis to Improve Portfolio Selection[J/OL]. The Journal of Business, 1973, 46(1): 66-86 [2023-10-05]. <http://www.jstor.org/stable/2351280>.
- [30] Goodwin T H. The Information Ratio[J/OL]. Financial Analysts Journal, 1998, 54(4): 34-43 [2023-10-05]. <http://www.jstor.org/stable/4480091>.
- [31] Jensen M C. THE PERFORMANCE OF MUTUAL FUNDS IN THE PERIOD 1945 – 1964[J/OL]. The Journal of Finance, 1968, 23(2): 389-416. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1968.tb00815.x>. DOI: <https://doi.org/10.1111/j.1540-6261.1968.tb00815.x>.
- [32] CHRISTIE W G, SCHULTZ P H. Why do NASDAQ Market Makers Avoid Odd-Eighth Quotes?[J/OL]. The Journal of Finance, 1994, 49(5): 1813-1840. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1994.tb04782.x>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1994.tb04782.x>. DOI: <https://doi.org/10.1111/j.1540-6261.1994.tb04782.x>.
- [33] CHRISTIE W G, HARRIS J H, SCHULTZ P H. Why Did NASDAQ Market Makers Stop Avoiding Odd-Eighth Quotes?[J/OL]. The Journal of Finance, 1994, 49(5): 1841-1860. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1994.tb04783.x>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1994.tb04783.x>. DOI: <https://doi.org/10.1111/j.1540-6261.1994.tb04783.x>.
- [34] O'Hara M. High frequency market microstructure[J/OL]. Journal of Financial Economics, 2015, 116(2): 257-270. <https://www.sciencedirect.com/science/article/pii/S0304405X15000045>. DOI: <https://doi.org/10.1016/j.jfineco.2015.01.003>.
- [35] Foucault T. Algorithmic Trading: Issues and Preliminary Evidence[M/OL]. in: Market Microstructure. John Wiley & Sons, Ltd, 2012: 1-40. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118673553.ch1>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118673553.ch1>. DOI: <https://doi.org/10.1002/9781118673553.ch1>.
- [36] Ammar I B, Hellara S, Ghadhab I. High-frequency trading and stock liquidity: An intraday analysis[J/OL]. Research in International Business and Finance, 2020, 53: 101235. <https://doi.org/10.1016/j.ribaf.2020.101235>. DOI: 10.1016/j.ribaf.2020.101235.

- [37] HENDERSHOTT T, JONES C M, MENKVELD A J. Does Algorithmic Trading Improve Liquidity?[J/OL]. The Journal of Finance, 2011, 66(1): 1-33. <https://doi.org/10.1111/j.1540-6261.2010.01624.x>. DOI: 10.1111/j.1540-6261.2010.01624.x.
- [38] Scholtus M, van Dijk D, Frijns B. Speed, algorithmic trading, and market quality around macroeconomic news announcements[J/OL]. Journal of Banking & Finance, 2014, 38: 89-105. <https://www.sciencedirect.com/science/article/pii/S0378426613003841>. DOI: <https://doi.org/10.1016/j.jbankfin.2013.09.016>.
- [39] Arumugam D, Prasanna P K, Marathe R R. Do algorithmic traders exploit volatility?[J/OL]. Journal of Behavioral and Experimental Finance, 2023, 37: 100778. <https://www.sciencedirect.com/science/article/pii/S2214635022001009>. DOI: <https://doi.org/10.1016/j.jbef.2022.100778>.
- [40] Anagnostidis P, Fontaine P, Varsakelis C. Are high-frequency traders informed? [J/OL]. Economic Modelling, 2020, 93: 365-383. <https://doi.org/10.1016/j.econmod.2020.08.013>. DOI: 10.1016/j.econmod.2020.08.013.
- [41] KIRILENKO A, KYLE A S, SAMADI M, et al. The Flash Crash: High-Frequency Trading in an Electronic Market[J/OL]. The Journal of Finance, 2017, 72(3): 967-998. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/jofi.12498>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/jofi.12498>. DOI: <https://doi.org/10.1111/jofi.12498>.
- [42] O'Hara M. Market Microstructure Theory[M]. London, England: Blackwell, 1997.
- [43] Hasbrouck J. Empirical market microstructure[M]. New York, NY: Oxford University Press, 2007.
- [44] Easley D. High-frequency Trading: New Realities for Traders, Markets and Regulators[M/OL]. Risk Books, 2013. <https://books.google.com.hk/books?id=pli5oAEACAAJ>.
- [45] Guo M, Li Z, Tu Z. A unique “T+1 trading rule” in China: Theory and evidence [J]. Journal of Banking & Finance, 2012, 36(2): 575-583. DOI: 10.1016/J.JBANFIN.2011.09.002.
- [46] Lopez de Prado M. Advances in financial machine learning[M]. Nashville, TN: John Wiley & Sons, 2018.

- [47] Kyle A S. Continuous Auctions and Insider Trading[J/OL]. *Econometrica*, 1985, 53(6): 1315. <http://dx.doi.org/10.2307/1913210>. DOI: 10.2307/1913210.
- [48] Amihud Y. Illiquidity and stock returns: cross-section and time-series effects [J/OL]. *Journal of Financial Markets*, 2002, 5(1): 31-56. [http://dx.doi.org/10.1016/S1386-4181\(01\)00024-6](http://dx.doi.org/10.1016/S1386-4181(01)00024-6). DOI: 10.1016/s1386-4181(01)00024-6.
- [49] EASLEY D, KIEFER N M, O'HARA M, et al. Liquidity, Information, and Infrequently Traded Stocks[J/OL]. *The Journal of Finance*, 1996, 51(4): 1405-1436. <http://dx.doi.org/10.1111/j.1540-6261.1996.tb04074.x>. DOI: 10.1111/j.1540-6261.1996.tb04074.x.
- [50] Easley D, Hvidkjaer S, O'Hara M. Is Information Risk a Determinant of Asset Returns?[J/OL]. *The Journal of Finance*, 2002, 57(5): 2185-2221. <http://dx.doi.org/10.1111/1540-6261.00493>. DOI: 10.1111/1540-6261.00493.
- [51] Loeb T F. Trading Cost: The Critical Link between Investment Information and Results[J/OL]. *Financial Analysts Journal*, 1983, 39(3): 39-44 [2023-10-10]. <http://www.jstor.org/stable/4478644>.
- [52] Arnott R D, Wagner W H. The Measurement and Control of Trading Costs [J/OL]. *Financial Analysts Journal*, 1990, 46(6): 73-80. eprint: <https://doi.org/10.2469/faj.v46.n6.73>. DOI: 10.2469/faj.v46.n6.73.
- [53] Wagner W H, Edwards M. Best Execution[J/OL]. *Financial Analysts Journal*, 1993, 49(1): 65-71. eprint: <https://doi.org/10.2469/faj.v49.n1.65>. DOI: 10.2469/faj.v49.n1.65.
- [54] Bertsimas D, Lo A W. Optimal control of execution costs[J/OL]. *Journal of Financial Markets*, 1998, 1(1): 1-50. [https://doi.org/10.1016/S1386-4181\(97\)00012-8](https://doi.org/10.1016/S1386-4181(97)00012-8). DOI: 10.1016/s1386-4181(97)00012-8.
- [55] Almgren R, Chriss N. Optimal execution of portfolio transactions[J/OL]. *The Journal of Risk*, 2001, 3(2): 5-39. <https://doi.org/10.21314/jor.2001.041>. DOI: 10.21314/jor.2001.041.
- [56] Freyre-Sanders A, Guobuzaitė R, Byrne K. A Review of Trading Cost Models [J/OL]. *The Journal of Investing*, 2004, 13(3): 93-115. <https://doi.org/10.3905/joi.2004.434557>. DOI: 10.3905/joi.2004.434557.

- [57] Rosenthal D. Performance Metrics for Algorithmic Traders[J]. SSRN Electronic Journal, 2009. DOI: 10.2139/ssrn.1439902.
- [58] Almgren R F. Optimal execution with nonlinear impact functions and trading-enhanced risk[J/OL]. Applied Mathematical Finance, 2003, 10(1): 1-18. <https://doi.org/10.1080/135048602100056>. DOI: 10.1080/135048602100056.
- [59] Lillo F, Farmer J D, Mantegna R N. Master curve for price-impact function [J/OL]. Nature, 2003, 421(6919): 129-130. <https://doi.org/10.1038/421129a>. DOI: 10.1038/421129a.
- [60] Potters M, Bouchaud J P. More statistical properties of order books and price impact[J/OL]. Physica A: Statistical Mechanics and its Applications, 2003, 324(1-2): 133-140. [https://doi.org/10.1016/s0378-4371\(02\)01896-4](https://doi.org/10.1016/s0378-4371(02)01896-4). DOI: 10.1016/s0378-4371(02)01896-4.
- [61] Farmer J D, Patelli P, Zovko I I. The predictive power of zero intelligence in financial markets[J/OL]. Proceedings of the National Academy of Sciences, 2005, 102(6): 2254-2259. <https://doi.org/10.1073/pnas.0409157102>. DOI: 10.1073/pnas.0409157102.
- [62] Moro E, Vicente J, Moyano L G, et al. Market impact and trading profile of hidden orders in stock markets[J/OL]. Physical Review E, 2009, 80(6). <https://doi.org/10.1103/physreve.80.066102>. DOI: 10.1103/physreve.80.066102.
- [63] Bacry E, Iuga A, Lasnier M, et al. Market Impacts and the Life Cycle of Investors Orders[J/OL]. Market Microstructure and Liquidity, 2015, 01(02): 1550009. <https://doi.org/10.1142/s2382626615500094>. DOI: 10.1142/s2382626615500094.
- [64] Han S, Wu L, Cheng Y. Equity market impact modeling: an empirical analysis for the Chinese market[J/OL]. Journal of Risk, 2018, 20(6): 75-97. <https://doi.org/10.21314/jor.2018.391>. DOI: 10.21314/jor.2018.391.
- [65] Said E, Ayed A B H, Thillou D, et al. Market impact: a systematic study of the high frequency options market[J/OL]. Quantitative Finance, 2020, 21(1): 69-84. <https://doi.org/10.1080/14697688.2020.1791948>. DOI: 10.1080/14697688.2020.1791948.
- [66] Tóth B, Lempérière Y, Deremble C, et al. Anomalous Price Impact and the Critical Nature of Liquidity in Financial Markets[J/OL]. Physical Review X, 2011,

- 1(2). <https://doi.org/10.1103/physrevx.1.021006>. DOI: 10.1103/physrevx.1.021006.
- [67] Guéant O. Permanent market impact can be nonlinear[EB/OL]. arXiv. 2013. <https://arxiv.org/abs/1305.0413>.
- [68] Almgren R, Thum C, Hauptmann E, et al. Direct Estimation of Equity Market Impact[J]. RISK, 2005, 18.
- [69] Frazzini A, Israel R, Moskowitz T J. Trading Costs[J]. Capital Markets: Market Microstructure eJournal, 2018. DOI: <https://dx.doi.org/10.2139/ssrn.3229719>.
- [70] Perold A F. The implementation shortfall[Z]. 1988. DOI: 10.3905/jpm.1988.409150.
- [71] Kissell R L. Transaction Costs[G/OL]. in: Algorithmic Trading Methods. Elsevier, 2021: 57-97. <https://doi.org/10.1016/b978-0-12-815630-8.00003-x>. DOI: 10.1016/b978-0-12-815630-8.00003-x.
- [72] Kissell R. Transaction Cost Analysis[J/OL]. The Journal of Trading, 2008, 3(2): 29-37. <https://doi.org/10.3905/jot.2008.705630>. DOI: 10.3905/jot.2008.705630.
- [73] Konishi H. Optimal slice of a VWAP trade[J/OL]. Journal of Financial Markets, 2002, 5(2): 197-221. [https://doi.org/10.1016/s1386-4181\(01\)00023-4](https://doi.org/10.1016/s1386-4181(01)00023-4). DOI: 10.1016/s1386-4181(01)00023-4.
- [74] Gomber P, Lutat M, Wranik A. Flexible VWAP Executions in Electronic Trading [G/OL]. in: Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2008: 1-14. https://doi.org/10.1007/978-3-540-78550-7_1. DOI: 10.1007/978-3-540-78550-7_1.
- [75] Biakowski J, Darolles S, Fol G L. Improving VWAP strategies: A dynamic volume approach[J/OL]. Journal of Banking & Finance, 2008, 32(9): 1709-1722. <https://doi.org/10.1016/j.jbankfin.2007.09.023>. DOI: 10.1016/j.jbankfin.2007.09.023.
- [76] McCulloch J, Kazakov V. Mean Variance Optimal VWAP Trading[J/OL]. SSRN Electronic Journal, 2012. <https://doi.org/10.2139/ssrn.1803858>. DOI: 10.2139/ssrn.1803858.
- [77] Frei C, Westray N. OPTIMAL EXECUTION OF A VWAP ORDER: A STOCHASTIC CONTROL APPROACH[J/OL]. Mathematical Finance, 2013,

- 25(3): 612-639. <https://doi.org/10.1111/mafi.12048>. DOI: 10.1111/mafi.12048.
- [78] Mitchell D, Bialkowski J P, Tompaidis S. Optimal VWAP Tracking[J/OL]. SSRN Electronic Journal, 2013. <https://doi.org/10.2139/ssrn.2333916>. DOI: 10.2139/ssrn.2333916.
- [79] Busseti E, Boyd S. Volume Weighted Average Price Optimal Execution[EB/OL]. arXiv. 2015. <https://arxiv.org/abs/1509.08503>.
- [80] Cartea Á, Jaimungal S. A Closed-Form Execution Strategy to Target Volume Weighted Average Price[J/OL]. SIAM Journal on Financial Mathematics, 2016, 7(1): 760-785. <https://doi.org/10.1137/16m1058406>. DOI: 10.1137/16m1058406.
- [81] Barzykin A, Lillo F. Optimal VWAP execution under transient price impact [EB/OL]. arXiv. 2019. <https://arxiv.org/abs/1901.02327>.
- [82] Buti S, Rindi B, Werner I M. Dark pool trading strategies, market quality and welfare[J/OL]. Journal of Financial Economics, 2017, 124(2): 244-265. <https://doi.org/10.1016/j.jfineco.2016.02.002>. DOI: 10.1016/j.jfineco.2016.02.002.
- [83] Moinas S. Hidden Limit Orders and Liquidity in Limit Order Markets[J/OL]. SSRN Electronic Journal, 2005. <https://doi.org/10.2139/ssrn.676564>. DOI: 10.2139/ssrn.676564.
- [84] Gatev E, Goetzmann W, Rouwenhorst K G. Pairs Trading: Performance of a Relative Value Arbitrage Rule[R/OL]. 1999. <https://doi.org/10.3386/w7032>.
- [85] McCarthy J, Minsky M L, Rochester N, et al. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955[J/OL]. AI Magazine, 2006, 27(4): 12. <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904>. DOI: 10.1609/aimag.v27i4.1904.
- [86] Boser B E, Guyon I M, Vapnik V N. A Training Algorithm for Optimal Margin Classifiers[C/OL]. in: COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992: 144-152. <https://doi.org/10.1145/130385.130401>. DOI: 10.1145/130385.130401.
- [87] Cortes C, Vapnik V. Support-Vector Networks[J/OL]. Mach. Learn., 1995, 20(3): 273-297. <https://doi.org/10.1023/A:1022627411411>. DOI: 10.1023

- /A:1022627411411.
- [88] Quinlan J R. Induction of decision trees[J/OL]. Machine Learning, 1986, 1(1): 81-106. <https://doi.org/10.1007/BF00116251>. DOI: 10.1007/BF00116251.
 - [89] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities.[J/OL]. Proceedings of the National Academy of Sciences, 1982, 79(8): 2554-2558. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554>. <https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554>. DOI: 10.1073/pnas.79.8.2554.
 - [90] Schmidhuber J. Deep learning in neural networks: An overview[J/OL]. Neural Networks, 2015, 61: 85-117. <https://www.sciencedirect.com/science/article/pii/S0893608014002135>. DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>.
 - [91] Gu S, Kelly B, Xiu D. Empirical Asset Pricing via Machine Learning[J/OL]. The Review of Financial Studies, 2020, 33(5): 2223-2273. eprint: <https://academic.oup.com/rfs/article-pdf/33/5/2223/33209812/hhaa009.pdf>. <https://doi.org/10.1093/rfs/hhaa009>. DOI: 10.1093/rfs/hhaa009.
 - [92] Nagel S. Machine learning in asset pricing[M]. Princeton, NJ: Princeton University Press, 2021.
 - [93] Chen L, Pelger M, Zhu J. Deep Learning in Asset Pricing[J/OL]. Management Science, 2023, 0(0): null. eprint: <https://doi.org/10.1287/mnsc.2023.4695>. <https://doi.org/10.1287/mnsc.2023.4695>. DOI: 10.1287/mnsc.2023.4695.
 - [94] Pan S, Long S, Wang Y, et al. Nonlinear asset pricing in Chinese stock market: A deep learning approach[J/OL]. International Review of Financial Analysis, 2023, 87: 102627. <https://www.sciencedirect.com/science/article/pii/S1057521923001436>. DOI: <https://doi.org/10.1016/j.irfa.2023.102627>.
 - [95] Aitken M, Cumming D, Zhan F. High frequency trading and end-of-day price dislocation[J]. Journal of Banking & Finance, 2015, 59: 330-349. DOI: 10.1016/J.JBANKFIN.2015.06.011.
 - [96] Ponomarev E S, Oseledets I V, Cichocki A S. Using Reinforcement Learning in the Algorithmic Trading Problem[J/OL]. Journal of Communications Technology and Electronics, 2019, 64(12): 1450-1457. <https://doi.org/10.1134/s1064226919120131>. DOI: 10.1134/s1064226919120131.

-
- [97] Bustos O, Pomares-Quimbaya A. Stock market movement forecast: A Systematic review[J/OL]. Expert Systems with Applications, 2020, 156: 113464. <https://www.sciencedirect.com/science/article/pii/S0957417420302888>. DOI: <https://doi.org/10.1016/j.eswa.2020.113464>.
 - [98] Soni P, Tewari Y, Krishnan D. Machine Learning Approaches in Stock Price Prediction: A Systematic Review[J/OL]. Journal of Physics: Conference Series, 2022, 2161(1): 012065. <https://dx.doi.org/10.1088/1742-6596/2161/1/012065>. DOI: 10.1088/1742-6596/2161/1/012065.
 - [99] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey[J/OL]. WIREs Data Mining and Knowledge Discovery, 2018, 8(4): e1253. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1253>. <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1253>. DOI: <https://doi.org/10.1002/widm.1253>.
 - [100] Yadav A, Vishwakarma D K. Sentiment analysis using deep learning architectures: a review[J/OL]. Artificial Intelligence Review, 2020, 53(6): 4335-4385. <https://doi.org/10.1007/s10462-019-09794-5>. DOI: 10.1007/s10462-019-09794-5.
 - [101] Swathi T, Kasiviswanath N, Rao A A. An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis[J/OL]. Applied Intelligence, 2022, 52(12): 13675-13688. <https://doi.org/10.1007/s10489-022-03175-2>. DOI: 10.1007/s10489-022-03175-2.
 - [102] Liu Q, Wang C, Zhang P, et al. Detecting stock market manipulation via machine learning: Evidence from China Securities Regulatory Commission punishment cases[J/OL]. International Review of Financial Analysis, 2021, 78: 101887. <https://www.sciencedirect.com/science/article/pii/S1057521921002143>. DOI: <https://doi.org/10.1016/j.irfa.2021.101887>.
 - [103] FISHER R A. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS[J/OL]. Annals of Eugenics, 1936, 7(2): 179-188. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>. DOI: 10.1111/j.1469-1809.1936.tb02137.x.
 - [104] Smola A J, Schölkopf B. A tutorial on support vector regression[J/OL]. Statistics and Computing, 2004, 14(3): 199-222. <https://doi.org/10.1023/b:stco.0000035301.49549.88>. DOI: 10.1023/b:stco.0000035301.49549.88.

-
- [105] Karush W. Minima of functions of several variables with inequalities as side conditions.[D]. Thesis (S.M.)—University of Chicago, Department of Mathematics, December 1939., 1939.
- [106] Kuhn H W, Tucker A W. Nonlinear Programming[J/OL]. Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (July 31 - August 12, 1950), 1950: pages 481-492. <http://digicoll.lib.berkeley.edu/record/112776>.
- [107] Awad M, Khanna R. Support Vector Machines for Classification[G/OL]. in: Efficient Learning Machines. Apress, 2015: 39-66. https://doi.org/10.1007/978-1-4302-5990-9_3. DOI: 10.1007/978-1-4302-5990-9_3.
- [108] Devroye L, Györfi L, Lugosi G. A Probabilistic Theory of Pattern Recognition [M/OL]. Springer New York, 1996. <https://doi.org/10.1007/978-1-4612-0711-5>. DOI: 10.1007/978-1-4612-0711-5.
- [109] Breiman L, Friedman J H, Olshen R A, et al. Classification And Regression Trees [M/OL]. Routledge, 2017. <https://doi.org/10.1201/9781315139470>. DOI: 10.1201/9781315139470.
- [110] Alpaydin E. Introduction to Machine Learning[M]. 4th ed. London, England: MIT Press, 2020.
- [111] LeCun Y, Bengio Y, Hinton G. Deep learning[J/OL]. Nature, 2015, 521(7553): 436-444. <https://doi.org/10.1038/nature14539>. DOI: 10.1038/nature14539.
- [112] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J/OL]. The Bulletin of Mathematical Biophysics, 1943, 5(4): 115-133. <https://doi.org/10.1007/bf02478259>. DOI: 10.1007/bf02478259.
- [113] Rosenblatt F. The Perceptron: A Perceiving and Recognizing Automaton[Z]. 1957.
- [114] KELLEY H J. Gradient Theory of Optimal Flight Paths[J/OL]. ARS Journal, 1960, 30(10): 947-954. <https://doi.org/10.2514/8.5282>. DOI: 10.2514/8.5282.
- [115] Minsky M, Papert S A. Perceptrons[M/OL]. The MIT Press, 2017. <https://doi.org/10.7551/mitpress/11301.001.0001>. DOI: 10.7551/mitpress/11301.001.0001.
- [116] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J/OL]. Nature, 1986, 323(6088): 533-536. <https://doi.org/10>

- .1038/323533a0. DOI: 10.1038/323533a0.
- [117] LeCun Y, Boser B, Denker J S, et al. Backpropagation Applied to Handwritten Zip Code Recognition[J/OL]. *Neural Computation*, 1989, 1(4): 541-551. <https://doi.org/10.1162/neco.1989.1.4.541>. DOI: 10.1162/neco.1989.1.4.541.
 - [118] Cybenko G. Approximation by superpositions of a sigmoidal function[J/OL]. *Mathematics of Control, Signals, and Systems*, 1989, 2(4): 303-314. <https://doi.org/10.1007/bf02551274>. DOI: 10.1007/bf02551274.
 - [119] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets[J/OL]. *Neural Computation*, 2006, 18(7): 1527-1554. <https://doi.org/10.1162/neco.2006.18.7.1527>. DOI: 10.1162/neco.2006.18.7.1527.
 - [120] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C/OL]. in: Gordon G, Dunson D, Dudík M. *Proceedings of Machine Learning Research: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*: vol. 15. Fort Lauderdale, FL, USA: PMLR, 2011: 315-323. <https://proceedings.mlr.press/v15/glorot11a.html>.
 - [121] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Networks[J/OL]. *Communications of the ACM*, 2014, 63(11): 139-144. arXiv: 1406.2661. <https://arxiv.org/abs/1406.2661v1>.
 - [122] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[EB/OL]. arXiv. 2017. <https://arxiv.org/abs/1706.03762>.
 - [123] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[EB/OL]. 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
 - [124] Brown T B, Mann B, Ryder N, et al. Language Models are Few-Shot Learners [EB/OL]. arXiv. 2020. <https://arxiv.org/abs/2005.14165>.
 - [125] OpenAI. GPT-4 Technical Report[EB/OL]. arXiv. 2023. <https://arxiv.org/abs/2303.08774>.
 - [126] Sarker I H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions[J/OL]. *SN Computer Science*, 2021, 2(6). <https://doi.org/10.1007/s42979-021-00815-1>. DOI: 10.1007/s42979-021-00815-1.

-
- [127] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-Learn: Machine Learning in Python[J]. J. Mach. Learn. Res., 2011, 12(null): 2825-2830.
- [128] Hinton G. The Forward-Forward Algorithm: Some Preliminary Investigations [EB/OL]. arXiv. 2022. <https://arxiv.org/abs/2212.13345>.
- [129] Elman J L. Finding Structure in Time[J/OL]. Cognitive Science, 1990, 14(2): 179-211. https://doi.org/10.1207/s15516709cog1402_1. DOI: 10.1207/s15516709cog1402_1.
- [130] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J/OL]. Neural Computation, 1997, 9(8): 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>. DOI: 10.1162/neco.1997.9.8.1735.
- [131] Cho K, van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[EB/OL]. arXiv. 2014. <https://arxiv.org/abs/1406.1078>.
- [132] Mirza M, Osindero S. Conditional Generative Adversarial Nets[J/OL]. 2014. arXiv: 1411.1784. <https://arxiv.org/abs/1411.1784v1>.
- [133] Odena A, Olah C, Shlens J. Conditional Image Synthesis With Auxiliary Classifier GANs[EB/OL]. arXiv. 2016. <https://arxiv.org/abs/1610.09585>.
- [134] Isola P, Zhu J Y, Zhou T, et al. Image-to-Image Translation with Conditional Adversarial Networks[EB/OL]. arXiv. 2016. <https://arxiv.org/abs/1611.07004>.
- [135] Littman M L. Reinforcement learning improves behaviour from evaluative feedback[J/OL]. Nature, 2015, 521(7553): 445-451. <https://doi.org/10.1038/nature14540>. DOI: 10.1038/nature14540.
- [136] Sutton R S, Barto A G. Reinforcement Learning[M]. 2nd ed. Cambridge, MA: Bradford Books, 2018.
- [137] Watkins C J C H, Dayan P. Q-learning[J/OL]. Machine Learning, 1992, 8(3-4): 279-292. <https://doi.org/10.1007/bf00992698>. DOI: 10.1007/bf00992698.
- [138] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[EB/OL]. arXiv. 2013. <https://arxiv.org/abs/1312.5602>.
- [139] Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms[EB/OL]. arXiv. 2017. <https://arxiv.org/abs/1707.06347>.

- [140] Breiman L. Bagging Predictors[J/OL]. Machine Learning, 1996, 24(2): 123-140. <https://doi.org/10.1023/a:1018054314350>. DOI: 10.1023/a:1018054314350.
- [141] Zhou Z H. Ensemble Methods[M/OL]. Chapman, 2012. <https://doi.org/10.1201/b12207>. DOI: 10.1201/b12207.
- [142] Sagi O, Rokach L. Ensemble learning: A survey[J/OL]. WIREs Data Mining and Knowledge Discovery, 2018, 8(4): e1249. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1249>. <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249>. DOI: <https://doi.org/10.1002/widm.1249>.
- [143] Dietterich T G. Ensemble Methods in Machine Learning[G/OL]. in: Multiple Classifier Systems. Springer Berlin Heidelberg, 2000: 1-15. https://doi.org/10.1007/3-540-45014-9_1. DOI: 10.1007/3-540-45014-9_1.
- [144] Breiman L. Random forests[J/OL]. Machine Learning, 2001, 45(1): 5-32. <https://doi.org/10.1023/A:1010933404324>. DOI: 10.1023/A:1010933404324.
- [145] Biau G, Scornet E. A random forest guided tour[J/OL]. TEST, 2016, 25(2): 197-227. <https://doi.org/10.1007/s11749-016-0481-7>. DOI: 10.1007/s11749-016-0481-7.
- [146] Freund Y, Schapire R E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting[J/OL]. Journal of Computer and System Sciences, 1997, 55(1): 119-139. <https://doi.org/10.1006/jcss.1997.1504>. DOI: 10.1006/jcss.1997.1504.
- [147] Friedman J H. Greedy function approximation: A gradient boosting machine. [J/OL]. The Annals of Statistics, 2001, 29(5). <https://doi.org/10.1214/aos/1013203451>. DOI: 10.1214/aos/1013203451.
- [148] Friedman J H. Stochastic gradient boosting[J/OL]. Computational Statistics & Data Analysis, 2002, 38(4): 367-378. [https://doi.org/10.1016/s0167-9473\(01\)00065-2](https://doi.org/10.1016/s0167-9473(01)00065-2). DOI: 10.1016/s0167-9473(01)00065-2.
- [149] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J/OL]. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 13-17-August-2016: 785-794. arXiv: 1603.02754. <https://arxiv.org/abs/1603.02754v3>. DOI: 10.1145/2939672.2939785.

-
- [150] Wolpert D H. Stacked generalization[J/OL]. *Neural Networks*, 1992, 5(2): 241-259. [https://doi.org/10.1016/s0893-6080\(05\)80023-1](https://doi.org/10.1016/s0893-6080(05)80023-1). DOI: 10.1016/s0893-6080(05)80023-1.
- [151] Roşu I. A Dynamic Model of the Limit Order Book[J/OL]. *Review of Financial Studies*, 2009, 22(11): 4601-4641. <http://dx.doi.org/10.1093/rfs/hhp011>. DOI: 10.1093/rfs/hhp011.
- [152] Huang R, Polak T. LOBSTER: Limit Order Book Reconstruction System[J/OL]. *SSRN Electronic Journal*, 2011. <http://dx.doi.org/10.2139/ssrn.1977207>. DOI: 10.2139/ssrn.1977207.
- [153] Abergel F, Jedidi A. A Mathematical Approach to Order Book Modelling [M/OL]. in: *Econophysics of Order-driven Markets*. Springer Milan, 2011: 93-107. http://dx.doi.org/10.1007/978-88-470-1766-5_7. DOI: 10.1007/978-88-470-1766-5_7.
- [154] Huang W, Lehalle C A, Rosenbaum M. Simulating and Analyzing Order Book Data: The Queue-Reactive Model[J/OL]. *Journal of the American Statistical Association*, 2015, 110(509): 107-122. <http://dx.doi.org/10.1080/01621459.2014.982278>. DOI: 10.1080/01621459.2014.982278.
- [155] Morariu-Patrichi M, Pakkanen M S. State-dependent Hawkes processes and their application to limit order book modelling[J/OL]. *Quantitative Finance*, 2021, 22(3): 563-583. <http://dx.doi.org/10.1080/14697688.2021.1983199>. DOI: 10.1080/14697688.2021.1983199.
- [156] Farmer J D, Foley D. The economy needs agent-based modelling[J/OL]. *Nature*, 2009, 460(7256): 685-686. <https://doi.org/10.1038/460685a>. DOI: 10.1038/460685a.
- [157] Shi Z, Cartlidge J. Neural Stochastic Agent-Based Limit Order Book Simulation: A Hybrid Methodology[EB/OL]. *arXiv*. 2023. <https://arxiv.org/abs/2303.00080>.
- [158] Takahashi S, Chen Y, Tanaka-Ishii K. Modeling financial time-series with generative adversarial networks[J/OL]. *Physica A: Statistical Mechanics and its Applications*, 2019, 527: 121261. <http://dx.doi.org/10.1016/j.physa.2019.121261>. DOI: 10.1016/j.physa.2019.121261.

- [159] Li J, Wang X, Lin Y, et al. Generating Realistic Stock Market Order Streams [J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(01): 727-734. <http://dx.doi.org/10.1609/aaai.v34i01.5415>. DOI: 10.1609/aaai.v34i01.5415.
- [160] Coletta A, Moulin A, Vyetrenko S, et al. Learning to Simulate Realistic Limit Order Book Markets from Data as a World Agent[C/OL]. in: ICAIF '22: Proceedings of the Third ACM International Conference on AI in Finance. New York, NY, USA: Association for Computing Machinery, 2022: 428-436. <https://doi.org/10.1145/3533271.3561753>. DOI: 10.1145/3533271.3561753.
- [161] Cont R, Cucuringu M, Kochems J, et al. Limit Order Book Simulation with Generative Adversarial Networks[J/OL]. SSRN Electronic Journal, 2023. <http://dx.doi.org/10.2139/ssrn.4512356>. DOI: 10.2139/ssrn.4512356.
- [162] Briola A, Turiel J, Aste T. Deep Learning modeling of Limit Order Book: a comparative perspective[J]. 2020.
- [163] Rajihy Y, Nermend K, Alsakaa A. Back-propagation artificial neural networks in stock market forecasting. an application to the Warsaw stock exchange WIG20 [J]. Aestimatio: The IEB International Journal of Finance, 2017(15): 88-99.
- [164] Hu H, Tang L, Zhang S, et al. Predicting the direction of stock markets using optimized neural networks with Google Trends[J]. Neurocomputing, 2018, 285: 188-195.
- [165] Shen J, Shafiq M O. Short-term stock market price trend prediction using a comprehensive deep learning system[J/OL]. Journal of Big Data, 2020, 7(1). <https://doi.org/10.1186/s40537-020-00333-6>. DOI: 10.1186/s40537-020-00333-6.
- [166] Zhang K, Zhong G, Dong J, et al. Stock Market Prediction Based on Generative Adversarial Network[J/OL]. Procedia Computer Science, 2019, 147: 400-406. <http://dx.doi.org/10.1016/j.procs.2019.01.256>. DOI: 10.1016/j.procs.2019.01.256.
- [167] Diqi M, Hiswati M E, Nur A S. StockGAN: robust stock price prediction using GAN algorithm[J/OL]. International Journal of Information Technology, 2022, 14(5): 2309-2315. <http://dx.doi.org/10.1007/s41870-022-00929-6>. DOI: 10.1007/s41870-022-00929-6.

-
- [168] Tsantekidis A, Passalis N, Tefas A, et al. Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks[C/OL]. in: 2017 IEEE 19th Conference on Business Informatics (CBI). IEEE, 2017. <http://dx.doi.org/10.1109/CBI.2017.23>. DOI: 10.1109/cbi.2017.23.
- [169] Gunduz H, Yaslan Y, Cataltepe Z. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations[J]. Knowledge-Based Systems, 2017, 137: 138-148.
- [170] Hoseinzade E, Haratizadeh S. CNNpred: CNN-based stock market prediction using a diverse set of variables[J]. Expert Systems with Applications, 2019, 129: 273-285.
- [171] Cao J, Wang J. Stock price forecasting model based on modified convolution neural network and financial time series analysis[J]. International Journal of Communication Systems, 2019, 32(12): e3987.
- [172] Lu W, Li J, Li Y, et al. A CNN-LSTM-based model to forecast stock prices[J]. Complexity, 2020, 2020: 1-10.
- [173] Ishwarappa, Anuradha J. Big data based stock trend prediction using deep CNN with reinforcement-LSTM model[J/OL]. International Journal of System Assurance Engineering and Management, 2021. <https://doi.org/10.1007/s13198-021-01074-2>. DOI: 10.1007/s13198-021-01074-2.
- [174] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory[J/OL]. PLOS ONE, 2017, 12(7): e0180944. <http://dx.doi.org/10.1371/journal.pone.0180944>. DOI: 10.1371/journal.pone.0180944.
- [175] Minh D L, Sadeghi-Niaraki A, Huy H D, et al. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network[J]. Ieee Access, 2018, 6: 55392-55404.
- [176] Baek Y, Kim H Y. ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module[J]. Expert Systems with Applications, 2018, 113: 457-480.
- [177] Shen J, Shafiq M O. Short-term stock market price trend prediction using a comprehensive deep learning system[J]. Journal of big Data, 2020, 7(1): 1-33.

- [178] Li Y, Bu H, Li J, et al. The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning[J]. *International Journal of Forecasting*, 2020, 36(4): 1541-1562.
- [179] Lee S W, Kim H Y. Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation [J]. *expert systems with applications*, 2020, 161: 113704.
- [180] Lee M C. Using support vector machine with a hybrid feature selection method to the stock trend prediction[J/OL]. *Expert Systems with Applications*, 2009, 36(8): 10896-10904. <https://doi.org/10.1016/j.eswa.2009.02.038>. DOI: 10.1016/j.eswa.2009.02.038.
- [181] Hu Z, Zhu J, Tse K. Stocks market prediction using Support Vector Machine [C/OL]. in: *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*. IEEE, 2013. <https://doi.org/10.1109/iciiii.2013.6703096>. DOI: 10.1109/iciiii.2013.6703096.
- [182] Rustam Z, Kintandani P, et al. Application of support vector regression in indonesian stock price prediction with feature selection using particle swarm optimisation[J]. *Modelling and Simulation in Engineering*, 2019, 2019.
- [183] Liagkouras K, Metaxiotis K. Stock Market Forecasting by Using Support Vector Machines[G/OL]. in: *Learning and Analytics in Intelligent Systems*. Springer International Publishing, 2020: 259-271. https://doi.org/10.1007/978-3-030-49724-8_11. DOI: 10.1007/978-3-030-49724-8_11.
- [184] Yuan X, Yuan J, Jiang T, et al. Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market[J]. *IEEE Access*, 2020, 8: 22672-22685.
- [185] Mahmoodi A, Hashemi L, Jasemi M, et al. A developed stock price forecasting model using support vector machine combined with metaheuristic algorithms [J/OL]. *OPSEARCH*, 2022, 60(1): 59-86. <https://doi.org/10.1007/s12597-022-00608-x>. DOI: 10.1007/s12597-022-00608-x.
- [186] Lai R K, Fan C Y, Huang W H, et al. Evolving and clustering fuzzy decision tree for financial time series data forecasting[J]. *Expert Systems with Applications*, 2009, 36(2): 3761-3773.

-
- [187] Zhou P Y, Chan K C, Ou C X. Corporate communication network and stock price movements: insights from data mining[J]. IEEE Transactions on Computational Social Systems, 2018, 5(2): 391-402.
- [188] Carta S M, Consoli S, Piras L, et al. Explainable machine learning exploiting news and domain-specific lexicon for stock market forecasting[J]. IEEE Access, 2021, 9: 30193-30205.
- [189] Jiang M, Liu J, Zhang L, et al. An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms[J]. Physica A: Statistical Mechanics and its Applications, 2020, 541: 122272.
- [190] Kim S, Ku S, Chang W, et al. Predicting the direction of US stock prices using effective transfer entropy and machine learning techniques[J]. IEEE Access, 2020, 8: 111660-111682.
- [191] Yun K K, Yoon S W, Won D. Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process[J]. Expert Systems with Applications, 2021, 186: 115716.
- [192] Almaafi A, Bajaba S, Alnori F. Stock price prediction using ARIMA versus XGBoost models: the case of the largest telecommunication company in the Middle East[J/OL]. International Journal of Information Technology, 2023, 15(4): 1813-1818. <http://dx.doi.org/10.1007/s41870-023-01260-4>. DOI: 10.1007/s41870-023-01260-4.
- [193] Huang C Y. Financial Trading as a Game: A Deep Reinforcement Learning Approach[EB/OL]. arXiv. 2018. <https://arxiv.org/abs/1807.02787>.
- [194] Pricope T V. Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review[EB/OL]. arXiv. 2021. <https://arxiv.org/abs/2106.00123>.
- [195] Zou J, Zhao Q, Jiao Y, et al. Stock Market Prediction via Deep Learning Techniques: A Survey[EB/OL]. arXiv. 2022. <https://arxiv.org/abs/2212.12717>.
- [196] Kumbure M M, Lohrmann C, Luukka P, et al. Machine learning techniques and data for stock market forecasting: A literature review[J/OL]. Expert Systems with Applications, 2022, 197: 116659. <http://dx.doi.org/10.1016/j.eswa.2022.116659>. DOI: 10.1016/j.eswa.2022.116659.

- [197] Patel M, Jariwala K, Chattopadhyay C. Deep Learning techniques for stock market forecasting: Recent trends and challenges[C/OL]. in: ICSIM 2023: 2023 The 6th International Conference on Software Engineering and Information Management. ACM, 2023. <http://dx.doi.org/10.1145/3584871.3584872>. DOI: 10.1145/3584871.3584872.
- [198] Agarwal B, Mittal N. Machine Learning Approach for Sentiment Analysis [G/OL]. in: Socio-Affective Computing. Springer International Publishing, 2015: 21-45. https://doi.org/10.1007/978-3-319-25343-5_3. DOI: 10.1007/978-3-319-25343-5_3.
- [199] Severyn A, Moschitti A. Twitter Sentiment Analysis with Deep Convolutional Neural Networks[C/OL]. in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015. <https://doi.org/10.1145/2766462.2767830>. DOI: 10.1145/2766462.2767830.
- [200] Landolt S, Wambsganss T, Söllner M. A Taxonomy for Deep Learning in Natural Language Processing[C/OL]. in: Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences, 2021. <https://doi.org/10.24251/hicss.2021.129>. DOI: 10.24251/hicss.2021.129.
- [201] Araci D. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models[Z]. 2019. arXiv: 1908.10063 [cs.CL].
- [202] Lopez-Lira A, Tang Y. Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models[Z]. 2023. arXiv: 2304.07619 [q-fin.ST].
- [203] Chullamonthon P, Tangamchit P. Ensemble of supervised and unsupervised deep neural networks for stock price manipulation detection[J/OL]. Expert Systems with Applications, 2023, 220: 119698. <http://dx.doi.org/10.1016/j.eswa.2023.119698>. DOI: 10.1016/j.eswa.2023.119698.
- [204] Craja P, Kim A, Lessmann S. Deep learning for detecting financial statement fraud[J/OL]. Decision Support Systems, 2020, 139: 113421. <http://dx.doi.org/10.1016/j.dss.2020.113421>. DOI: 10.1016/j.dss.2020.113421.
- [205] Ali A, Abd Razak S, Othman S H, et al. Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review[J/OL]. Applied Sciences,

-
- 2022, 12(19): 9637. <http://dx.doi.org/10.3390/app12199637>. DOI: 10.3390/app12199637.
- [206] Sun E W, Meinel T. A new wavelet-based denoising algorithm for high-frequency financial data mining[J/OL]. *European Journal of Operational Research*, 2012, 217(3): 589-599. <http://dx.doi.org/10.1016/j.ejor.2011.09.049>. DOI: 10.1016/j.ejor.2011.09.049.
- [207] Officer R R. The Distribution of Stock Returns[J/OL]. *Journal of the American Statistical Association*, 1972, 67(340): 807-812. <http://dx.doi.org/10.1080/01621459.1972.10481297>. DOI: 10.1080/01621459.1972.10481297.
- [208] Fukunaga T, Umeno K. Universal Lévy's stable law of stock market and its characterization[EB/OL]. arXiv. 2017. <https://arxiv.org/abs/1709.06279>.
- [209] Yan H, Han L. Empirical distributions of stock returns: Mixed normal or kernel density?[J/OL]. *Physica A: Statistical Mechanics and its Applications*, 2019, 514: 473-486. <http://dx.doi.org/10.1016/j.physa.2018.09.080>. DOI: 10.1016/j.physa.2018.09.080.
- [210] Kohonen T. The self-organizing map[J/OL]. *Neurocomputing*, 1998, 21(1-3): 1-6. [http://dx.doi.org/10.1016/S0925-2312\(98\)00030-7](http://dx.doi.org/10.1016/S0925-2312(98)00030-7). DOI: 10.1016/s0925-2312(98)00030-7.