

基于 Informer 股票趋势预测模型的量化投资策略

报告简介

团队构建了面向股票趋势预测的 Informer 模型，旨在预测未来股票走势，从而更好的做出决策。不同于其他面向收益率的传统止损策略，团队采取面向模型的止损策略，赋予模型更大的决策权。回测结果表明，Informer 具有良好的预测趋势能力。同时团队通过冻结参数方法，使用每日新增的数据对模型结构做出修正，从而提高预测精度。最后，团队还发现在分钟线上，Informer 具有更加强大的预测能力，这得益于时间戳处理机制，但是由于数据量过大，算力不足，并未有时间进行全面测试。在未来，团队将填补分钟线交易策略的空白，继续挖掘深度学习模型与量化投资的关系。

本量化投资策略报告主要由以下三个部分组成：1. Informer 股票趋势预测模型 2. 模型止损策略 3. 迁移学习及其参数冻结，具体如下：

Informer 股票趋势预测模型：使用 Informer 模型进行股票投资是基于其卓越的性能和高效性。Informer 模型通过自注意力机制和稀疏自注意力算法，能够有效处理长时间序列数据，显著提升预测精度和效率。此外，Informer 模型能够捕捉时间序列中的长期依赖关系，并在处理大规模数据时表现出色。为了防止过拟合并提升模型的预测准确度，团队在策略中集成了正则化策略和早停机制。正则化通过在损失函数中加入惩罚项，限制模型复杂度，减少过拟合风险。早停机制则在训练过程中监控验证集上的表现，及时停止训练以避免过拟合。本策略应用 Informer 模型预测股票日线，选择 10 支股票，每支股票各自训练一个独立的 Informer 模型进行预测，每日预测下一日的价格变化。这种方法不仅发挥了 Informer 模型在时间序列数据处理上的优势，还结合了科学的风险管理策略，旨在市场波动中实现稳健的投资收益。

ATR 与模型止损策略：我们构造了 ATR 止损指标，并以其为基础构建了止损线，当价格触及止损线时，触发止损操作，抛售股票。同时我们构造了模型的止

损策略，如果模型出现多次失误，将暂停使用该模型，并重新进行适量训练以寻找最优参数。当模型的预测准确率提升至设定的阈值后，将重新启用该模型进行投资。此策略确保在模型表现不佳时，及时调整并优化模型，从而提高整体投资的准确性和稳定性。

迁移学习及其参数冻结：由于股票数据实时更新，模型也需随之更新。但每次完全重新训练不仅会导致过拟合和性能下降，而且耗时长。为解决此问题，团队采用迁移学习中的参数冻结方法，避免过拟合。将训练后的 Informer 模型用于预测其他未知股票时，表现出比其他神经网络模型更强的预测性能，显示了其在处理不同股票和指数预测任务中的显著优势。回测结果：本报告通过遗传算法和交叉验证寻 Informer 模型的最优参数集合，使得策略收益最优。

回测结果：本报告通过反复调参，对全部模型进行回测，分别选取十年长期回测 2010-2020，近半年回测 2021 年 12 月 1 日-2022 年 6 月 1 日，回测结果收益率均良好，可以得出本策略有效。

创新及亮点：

1. 应用 Informer 模型：采用深度学习中 Informer 模型进行股票趋势预测，充分利用其在处理长时间序列数据中的优势。
2. 引入 ATR 止损指标，模型止损策略，若模型多次失误，将暂停使用并重新训练，确保模型在表现不佳时及时调整并优化，提升整体投资准确性和稳定性。
3. 采用参数冻结方法，动态学习最新的股票数据，从而减少训练时间并提高模型在不同股票和指数预测任务中的性能。

目录

1 量化投资策略思路	7
1.1 量化策略流程图	7
1.2 投资策略简介	7
2 文献综述	8
2.1 量化投资的发展	8
2.2 人工智能在量化投资中的应用	9
2.3 informer 模型在时间序列预测中的应用	9
3 基于 Informer 的多因子选股模型	9
3.1 Informer 算法简介	9
3.1.1 稀疏自注意力机制 (ProbSparseSelf-Attention)	10
3.1.2 生成式时间序列建模 (GenerativeStyleDecoder)	10
3.1.3 二阶段检测 (Two-StageDetection)	10
3.2 informer 原理推导	10
3.2.1 原 transformer 结构	11
3.2.2 稀疏自注意力机制 (ProbSparseattention)	11
3.2.3 自注意蒸馏操作	13
3.2.4 生成推理法	14
3.2.5 全局时间戳	14
3.3 Informer 模型构建	16
3.3.1 数据获取	16
3.3.2 特征和标签提取	17
3.3.3 特征预处理	17
3.3.4 训练集和交叉验证集的合成	17
3.3.5 模型训练	18

3.3.6 交叉验证调参	18
3.4Informer 模型设定和调参	18
3.4.1Informer 超参数说明	19
3.4.2 调参算法简介	19
3.4.3 模型设定和调参结果	20
3.5 模型评估	21
4. 风险控制	21
4.1 主要风险	21
4.2 ATR 止损策略	22
4.3 模型止损策略	23
5 迁移学习及参数冻结	23
5.1 预测误差问题	23
5.2 参数冻结	24
6 回测	26
6.1 2023 年回测	26
6.2 2022 年回测	26
6.3 2024 年回测	27
7 总结	28
8 参考文献	29

图目录

图 1.1	投资总策略	7
图 3.1	transform' Structurevsinformer' Structure	13
图 3.2	自注意蒸馏操作	13
图 3.3	transformer' s decoder vs Informer' s decoder	14
图 3.4	全局时间戳处理	15
图 3.5	流程图	16
图 3.6	遗传算法图示[6]	20
图 5.1	损失函数趋势图	24

表目录

表 3.1	Informer 超参数表	19
表 3.2	Training Parameters	20
表 5.1	参数冻结与参数未冻结时间对比实验	25
表 5.2	参数冻结与参数未冻结损失函数对比实验	25

1 量化投资策略思路

团队以 Informer 模型为核心构造量化投资策略思路。

1.1 量化策略流程图

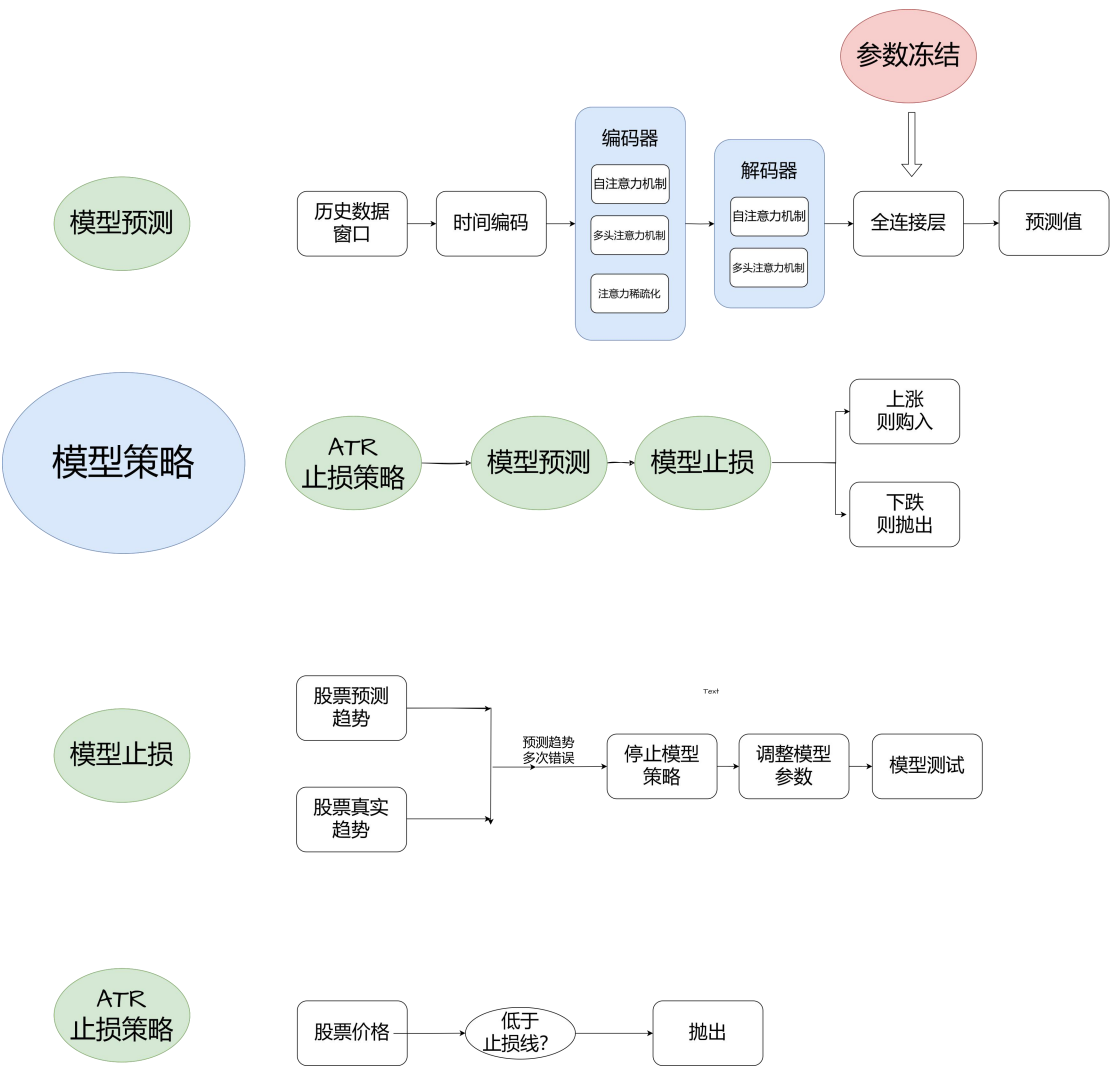


图 1.1 投资总策略

1.2 投资策略简介

传统的人工策略方法和机器学习方法在特征选取和特征工程构建上都极为复杂，并且高度依赖行业经验。然而，即使是经验丰富的专业人士，也难以避免特征选取和构造的失误。与此不同，深度学习模型无需进行复杂的特征工程，这是由于其多层次结构和强大的自动特征学习能力使得可以进行深度特征提取。

团队构造了 `get_attrs` 函数，在基于基础指标 'Date'、'Open'、'High'、'Low'、'Volume' 构造其余 53 个股票指标和一些相关参数，包括：'SMA'、'EMA'、'WMA'、'DEMA'、'TEMA'、'TRIMA'、'KAMA'、'MAMA'、'T3'、'ma5'、'ma30'、'ma_120'、'ma_250'、'Close'、'H_line'、'M_line'、'L_line'、'macd' 等，即移动平均线，指数移动平均线、布林带、平滑异同移动平均线、相对强弱指数、能量潮价格变换、周期指标、波动率指标、数学运算统计函数、数学变换等等。

将经过训练后的模型用于预测未来股票价值，并判断股票未来走势。基于模型的结果，我们决定是否买入股票。为了防止股市短期波动导致的过度亏损，我们构造 ATR 止损指标并构造了以 ATR 为基础的止损线；同时防止模型精度降低导致的过度亏损，我们构造了模型止损策略，确保模型精度高于阈值时才基于模型结果进行买入或卖出。

2 文献综述

2.1 量化投资的发展

量化投资是一种利用数学、统计学和计算机技术来做出投资决策的投资方式。其核心在于通过定量分析和模型构建，从历史数据中发现市场规律，制定并执行投资策略。20 世纪 70 年代，简化的数量模型和统计方法被首次应用于投资领域。马科维茨 (Harry Markowitz) 的现代投资组合理论和夏普 (William Sharpe) 的资本资产定价模型 (CAPM)^[1] 为量化投资奠定了理论基础。20 世纪 90 年代，随着计算机技术的进步和市场数据的丰富，算法交易 (Algorithmic Trading) 开始崭露头角。算法交易通过计算机程序自动执行交易策略，能够在短时间内处理

大量交易订单，提高了交易效率和精确度。21 世纪初，高频交易迅速发展。HFT 策略利用高速计算机和网络设施，在极短时间内完成交易，从微小的价格差异中获利。HFT 对市场流动性和定价效率产生了重要影响，但也引发了关于市场稳定性和公平性的讨论。

2.2 人工智能在量化投资中的应用

人工智能（AI）的引入为量化投资带来了革命性的变化。AI 技术，特别是机器学习和深度学习，能够处理复杂的非线性关系和大规模数据，为投资策略的开发和优化提供了强有力的工具。。Atsalakis 和 Valavanis（2009）应用 SVM 预测股市指数^[2]，证明了机器学习在市场预测中的潜力。深度学习近年来在金融市场预测中取得了显著成果。LongShort-TermMemory（LSTM）和 GatedRecurrentUnit（GRU）等递归神经网络（RNN）在时间序列预测中表现突出。Fischer 和 Krauss（2018）的研究表明^[3]，LSTM 在股票价格预测中的效果较优。

2.3 INFORMER 模型在时间序列预测中的应用

Informer 模型是专为长期序列数据预估需求开发的深度学习工具，其独特之处在于运用稀疏自注意力机制以高效地应对传统注意力方法面临的长序列数据运算效率与性能挑战。此稀疏特质能提升模型训练效率及预测准确率。此外，Informer 还能有效捕捉长时间依赖关系，避免传统 RNN 和 LSTM 在处理此类问题时可能产生的梯度消失现象。

实际应用中，Informer 已在电力负荷预测、气象预报以及金融时间序列预测等多个领域展现出卓越性能。团队创新地将 Informer 模型结合量化投资策略，期望实现更为精确的市场预测和投资决策。

3 基于 INFORMER 的多因子选股模型

3.1 INFORMER 算法简介

Informer (Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting) 模型是由 Haoyi、Zhou 等人在 2020 年提出的^[5], 旨在解决长时间序列预测中的效率和精度问题。作为一种创新的深度学习模型, Informer 基于 Transformer 架构进行了优化, 结合了稀疏自注意力机制和生成式时间序列建模。其设计目标在于通过高效计算和灵活处理, 显著提升长时间序列预测的性能。Informer 模型通过高效的长时间序列预测优化模型, 主要借助稀疏自注意力机制和生成式时间序列建模来实现模型的高效性和精确性。Informer 的核心创新包括稀疏自注意力机制、生成式解码器和二阶段检测模块。

3.1.1 稀疏自注意力机制 (PROBSPARSESELF-ATTENTION)

Informer 模型引入稀疏自注意力机制, 通过概率抽样和 Top-K 筛选来选择性地关注重要的时间点。这种机制在计算自注意力时, 根据概率分布选择子集进行计算, 而不是全量计算, 从而显著降低计算复杂度。Top-K 筛选进一步优化, 只保留对预测结果影响最大的前 K 个注意力值, 使模型在保证性能的前提下, 大幅提高计算效率。

3.1.2 生成式时间序列建模 (GENERATIVESTYLEDECODER)

Informer 采用生成式解码器, 直接生成未来一段时间的预测值, 避免了逐步解码带来的误差累积问题。生成式解码器通过直接输出整个预测序列, 提高了预测的稳定性和准确性。

3.1.3 二阶段检测 (TWO-STAGEDETECTION)

Informer 模型采用二阶段检测方法, 通过全局粗粒度检测和局部细粒度检测相结合, 捕捉序列中的重要模式和趋势。第一阶段进行全局检测, 提取整体特征; 第二阶段进行细粒度检测, 进一步捕捉局部细节特征, 从而提高模型对长时间序列数据的理解和预测能力。

3.2 INFORMER 原理推导

3.2.1 原 TRANSFORMER 结构

原始的 Transformer 模型主要由编码器和解码器组成^[4]，每个部分又包括多个层的堆叠。这些层包括自注意力机制和前馈神经网络。

编码器层（Encoder）：由多个相同的编码器组成，每层包括一个自注意力机制和一个前馈神经网络。

解码器层（Decoder）：由多个相同的解码器组成，每层包括一个自注意力机制、一个编码器-解码器注意力机制和一个前馈神经网络。

为了捕获序列关系，Transformer 中运用了自注意力机制，计算公式为：

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (1)$$

其中，Q、K、V 分别表示查询、键和值矩阵， d_k 是键的维度。

同时为了提高并行计算效率和加强特征表示，Transformer 模型采用多头注意力机制，通过并行计算多个自注意力，实现不同子空间的表示：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (2)$$

其中，每个 $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ 。

但是传统 Transformer 模型存在一定的局限，包括以下几个方面：

1. 每层的时间复杂度和内存使用量为 $O(L^2)$ ，其中 L 为序列的长度。
2. 长输入叠加层的内存瓶颈，限制了模型在接收长序列输入时的可扩展性。
3. 解码器的逐步推理使得预测长输出的速度大幅度下降。

为了解决上述三个问题，Informer 对 Transformer 增加了三个重要的变化和一些小的变化。Informer 和 Transformer 的主要结构如图 3.1 所示。

3.2.2 稀疏自注意力机制 (PROBSPARSEATTENTION)

在原 Transformer 中，计算输入，Query ($Q \in \mathbb{R}^{L_Q \times d}$)、Key ($K \in \mathbb{R}^{L_K \times d}$) 和 Value ($V \in \mathbb{R}^{L_V \times d}$) 的自注意力机制消耗了大量的计算资源。然而，在概率研究中，自我注意的概率分布表现出潜在的稀疏性^[5]，这表明大多数注意的作用相对较弱，只有少数点积配对对主要注意有显著影响。基于上述研究，Informer 提出稀疏自注意机制，将第 i 个查询的注意定义为概率形式：

$$A(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = E_{p(k_j|q_i)}[V_j] \quad (3)$$

式中 $p(k_j|q_i) = \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)}$ ，且 $k(q_i, k_j)$ 选择非对称指数核 $e^{\frac{q_i k_j}{\sqrt{d}}}$ 。为了衡量查询的稀疏性，使用 Kullback-Leibler 散度计算查询的注意概率分布与均匀分布概率分布的相对熵，其中第 i 次查询的稀疏性的评价公式为：

$$M(q_i, K) = \ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j}{\sqrt{d}} \quad (4)$$

其中 $\ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j}{\sqrt{d}}}$ ，为 q_i 在所有 Key 上的 $\frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j}{\sqrt{d}}$ 为算术平均值。

最终的 ProbSparse 自注意力公式为：

$$A(Q, K, V) = \text{Softmax} \left(\frac{\bar{Q}K}{\sqrt{d}} \right) V \quad (5)$$

其中， \bar{Q} 是与 Q 大小相同的稀疏矩阵，它只包含稀疏度量 $M(q, K)$ 。

通过以上步骤，可以消除”Lazy” Query，保持”Active” Query；因此，空间复杂度从 $O(L^2)$ 降低到 $O(L \ln L)$ 。有效地解决了原网络中二次元的计算复杂度，极大提高 Informer 训练速度。

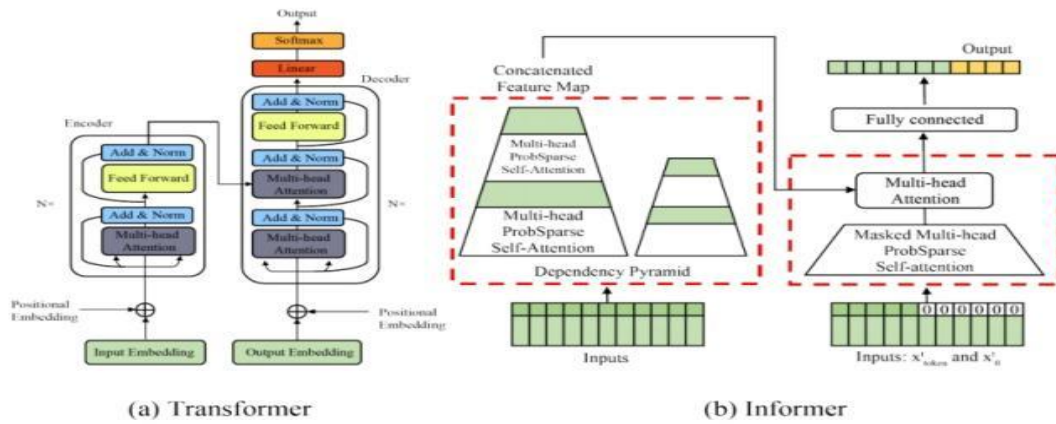


图 3.1 transform' Structurevsinformer' Structure

3.2.3 自注意蒸馏操作

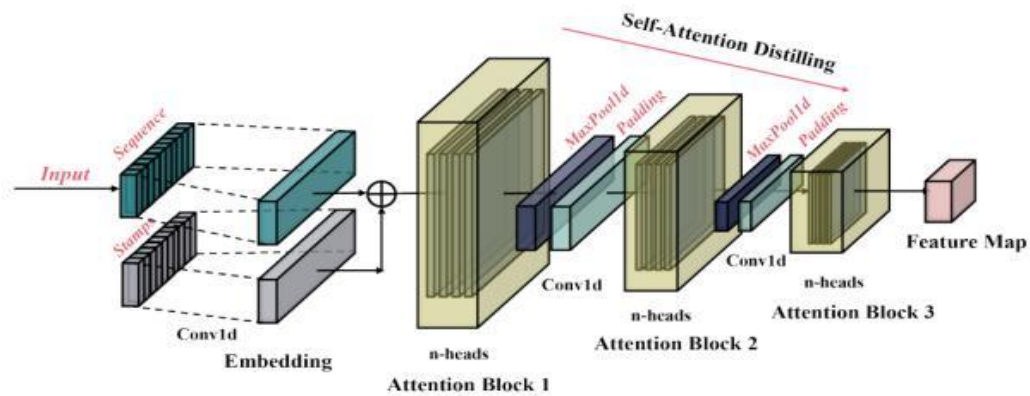


图 3.2 自注意蒸馏操作

在编码器部分进行了自注意提取操作。编码器是为了未来提取长序列输入的依赖关系而设置的结构，这也是 ProbSparse 自注意力机制的结果。编码器的特征映射具有冗余的值组合。因此，使用提取操作是为具有主导特征的主导特征分配更高的权重，并在下一层生成焦点自注意特征映射。从第 j 层到第 $j+1$ 层的蒸馏操作为：

$$X_{j+1}^t = \text{Maxpool}(\text{ELU}(\text{Conv1d}([X_j^t]_{AB}))), \quad (6)$$

其中 $[X_{j+1}^t]_{AB}$ 包含注意块和多头中的关键操作 ProbSparse 自注意力机制；Conv1d 表示对时间序列的一维卷积操作；ELU 是激活函数。蒸馏机制允许解码器的每一层将输入序列的长度减少一半，从而大大节省了存储成本和计算时间。详细的蒸馏过程如图 3.2 所示。

3.2.4 生成推理法

为了应对长序列预测问题，基于标准的解码器结构，采用生成式推理方法来减少长期预测的速度下降。解码器输入以下矢量：

$$X_{de}^t = \text{Concat}(X_{token}^t, X_0^t) \in R^{(L_{token}+L_y) \times d_{model}}, \quad (7)$$

其中， $X_{token}^t \in R^{L_{token} \times d_{model}}$ 为已知导向序列， $X_0^t \in R^{L_y \times d_{model}}$ 为待预测序列。这种解码器样式可以一次输出所有 X_0^t 。图 3.3 基于已知引导序列的 Informer 的精细化过程；而不是像基于 RNN 的模型那样动态而低效的级联。这种新型的推断和输出是实时的，更直观的，使其更容易随后构建交易策略。传统变压器和变换器的输出部分如图 3.3 所示。

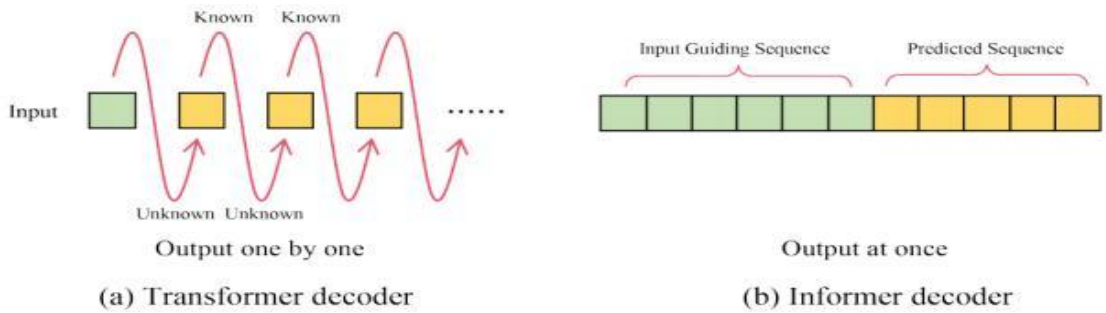


图 3.3 transformer' s decoder vs Informer' s decoder

3.2.5 全局时间戳

Transformer 采用多头自注意机制，避免了 RNN 的递归方法，加快了训练时间。然而，当每个序列同时通过 Transformer 的编码器和解码器堆栈时，模型本身没有任何位置感。因此，需要采用位置嵌入的方法将输入的顺序信息纳入模型。在位置嵌入的基础上，Informer 引入了全局时间戳嵌入操作，使时间戳也成为特征之一。时间戳部分包括年、月、周、日。通过嵌入时间戳，该模型能够准确匹配使用 Informer 网络数据到某一天，给每个序列一个独特的时间特征。嵌入后的序列被送入解码器。具体处理如图 3.4 所示。

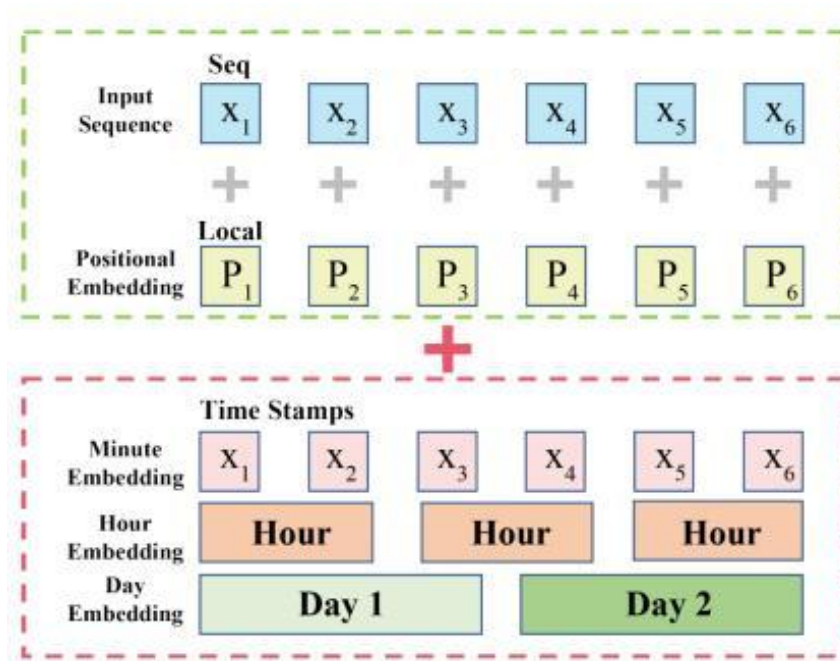


图 3.4 全局时间戳处理

与典型的关注长期股票变化的研究不同，日水平的成交量变化较为不平稳，本策略关注的是日线水平股票波动的动态预测，而日线水平的成交量波动有时剧烈。由于成交量的突然增加，模型可能存在梯度爆炸的风险，也可能被模型视为极值，显著影响股价预测，导致局部误差增大。然而，每个交易日开始和结束时的成交量决定了全天的基调以及下一个交易日的趋势。此外，不稳定的交易量经常导致股票价格的急剧变化。做出准确的预测可以减少无数的财务风险，带来巨

大的盈利机会。通过在打开和关闭数据上放置特殊的时间戳，模型可以对这些数据进行不同的处理，从而避免不必要的错误。

3.3 INFORMER 模型构建

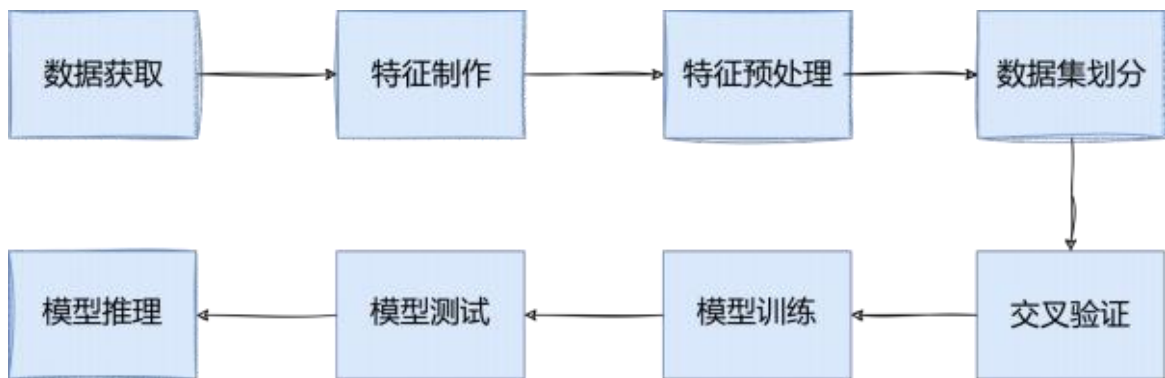


图 3.5 流程图

3.3.1 数据获取

为了验证模型的鲁棒性，选取了十只股票：000001.SZ（平安银行）、000002.SZ（万科A）、000039.SZ（中集集团）、000063.SZ（中兴通讯）、000338.SZ（潍柴动力）、000703.SZ（恒逸石化）、000858.SZ（五粮液）、002024.SZ（苏宁易购）、002142.SZ（宁波银行）、002594.SZ（比亚迪）、000012.SZ

同时为了防止数据泄露，不同测试集，设置了不同的训练区间和测试区间：对于测试区间为 2020-2023，设置训练区间为 2002 年 1 月 1 日至 2020 年 1 月 1 日；对于测试区间为 2022-2023，设置训练区间为 2002 年 1 月 1 日至 2022 年 1 月 1 日；对于测试区间为 2018-2023，设置训练区间为 2002 年 1 月 1 日至 2018 年 1 月 1 日。

3.3.2 特征和标签提取

团队基于股票的开盘价、最低价、最高价、收盘价以及交易量，手动构建了另外 52 个股票指标。采用滑动窗口划分技术，将每 64 天的数据作为特征，预测长度为 1 天，即预测下一日的收盘价。

3.3.3 特征预处理

在使用 Informer 模型之前，需要对特征进行预处理，以确保数据质量和模型效果。预处理包括以下几个步骤：

1. 时间特征处理：将时间编码为模型可识别的形式，将日期转换为时间戳。
2. 缺失值处理：对于缺失值，团队采用填充法进行处理，确保数据的完整性。
3. 标准化：为了加速模型训练并提高收敛性，所有特征都进行标准化处理。即将特征值都缩放到均值为 0、标准差为 1 的标准正态分布。

3.3.4 训练集和交叉验证集的合成

为了确保模型的泛化能力和鲁棒性，数据被划分为训练集和交叉验证集，并对其进行合成处理。

根据之前设定的时间区间，将数据集划分为训练集和验证集。例如，对于测试区间 2020-2023 年，将 2002-2020 年的数据作为训练集。对于每个训练区间，将数据划分为若干子集，保证训练集和验证集的时间跨度覆盖不同市场条件，增强模型的泛化能力。采用滑动窗口技术，将训练集数据进一步划分为多个重叠的子序列，每个子序列长度为 64 天。这样可以生成多组训练样本，提高模型的稳定性和鲁棒性。最后使用 5 折交叉验证方法，将训练集进一步划分为 5 个子集。在每次验证中，选择其中一个子集作为验证集，其余子集作为训练集。循环 5 次，确保每个子集都被作为验证集进行验证，从而获得模型在不同数据上的表现。

通过以上步骤，生成了高质量的训练集和交叉验证集，为后续的模型训练和验证提供了可靠的数据基础。这样可以更好地评估模型的性能，并确保其在实际应用中的有效性。

3.3.5 模型训练

在完成特征提取和预处理后，Informer 进入模型的训练阶段。

1. 模型架构

输入层：接受预处理后的特征数据。每个输入样本包含 64 天的特征数据。

编码器：基于稀疏注意力机制的编码器，对输入序列进行特征提取和表示。

解码器：通过解码器生成预测结果，即下一日的收盘价。

全连接层：将解码器的输出通过全连接层转换为最终的预测值。

2. 训练过程

在使用 Informer 模型进行时间序列预测的过程中，首先根据预处理后的特征数据初始化模型参数。接下来，进行数据加载和批处理，将训练数据按批次加载到模型中，逐批进行训练。在每个训练批次中，通过前向传播将输入特征数据传递给模型，生成预测值。随后，使用损失函数计算模型预测值与真实值之间的误差。在计算出误差后，进行反向传播，通过反向传播算法计算梯度，并使用优化器更新模型参数。

在每个训练周期后，使用交叉验证集对模型进行评估，根据验证集上的表现决定是否提前停止训练，以防止过拟合。训练结束后，使用 PyTorch 提供的 `torch.save()` 函数保存最佳模型参数，以便在实际应用中使用。通过这一系列步骤，确保模型能够有效学习并在实际应用中提供准确的预测结果。

3.3.6 交叉验证调参

在样本内训练完成后，团队会使用模型对交叉验证集进行预测，然后选取模型的最优参数，即交叉验证集中损失函数最小的一组参数。

3.4 INFORMER 模型设定和调参

3.4.1 INFORMER 超参数说明

表 3.1 Informer 超参数表

超参数	超参数说明
Batchsize（批处理大小）	更新参数的样本数量
Lossfunction（损失函数）	预测结果与实际值之间差异的函数。
Inputsequence（输入序列）	模型用于预测的输入序列的长度
Guidingsequence（引导序列）	引导模型进行更准确预测的序列长度。
Predictedsequence（预测序列）	预测的未来时间步数。
Epoch（训练轮数）	训练集上进行训练的次数。
e__layers（编码器层数）	编码器中 Transformer 层的数量。
d__layers（解码器层数）	解码器中 Transformer 层的数量。
d__ff__（全连接层大小）	Transformer 模型中全连接层的大小。
dropout（丢弃率）	Dropout 层的丢弃率。
activation（激活函数）	用于激活函数的类型。
n__heads（注意力头数）	多头注意力机制中的头数。

3.4.2 调参算法简介

遗传算法是一种模拟自然选择和遗传机制的优化算法，通过模拟自然界中生物的进化过程，逐代进化出更优解的优化算法。遗传算法通常包括以下几个基本步骤：

1. 初始化种群：随机生成初始种群，种群中的每个个体为一个候选解。
2. 适应度评估：计算每个个体在目标函数上的表现。
3. 选择操作：根据个体的适应度值，选择一定数量个体，繁殖下一代。
4. 交叉操作：对选出的父代个体进行交叉操作，产生新的个体。

5. 变异操作：对新生成的个体进行变异操作，增加种群的多样性。
6. 更新种群：将原始种群和新生成的个体结合，形成下一代种群。
7. 终止条件：重复上述步骤，直到满足目标函数收敛。

Algorithm 1: *Classical Genetic Algorithm (GA)*

Input:
Population Size, n
Maximum number of iterations, MAX

Output:
Global best solution, Y_{bt}

begin
Generate initial population of n chromosomes Y_i ($i = 1, 2, \dots, n$)
Set iteration counter $t = 0$
Compute the fitness value of each chromosomes
while ($t < MAX$)
 Select a pair of chromosomes from initial population based on fitness
 Apply crossover operation on selected pair with crossover probability
 Apply mutation on the offspring with mutation probability
 Replace old population with newly generated population
 Increment the current iteration t by 1.
end while
return the best solution, Y_{bt}
end

图 3.6 遗传算法图示[6]

3.4.3 模型设定和调参结果

最终，通过手动调参、随机搜索算法、遗传算法等一系列调参手段，模型精度达到最高，最终选取参数结果如表 3.1 所示。

表 3.2 Training Parameters

Parameter	Value
Batchsize	32
Lossfunction	MSE
Inputsequence	64
Guidingsequence	32

Predictedsequence	1
Epoch	1000
e__layers	3
d__layers	2
d__ff	512
dropout	0.001
activation	GELU
n__heads	8

3.5 模型评估

对于时序问题，通常使用 MeanAbsoluteError(MAE) 作为损失函数进行评估。MAE 损失函数表示为真实曲线和预测曲线的误差。MAE 越小，预测准确率越高。

$$MAE = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{x}_i - x_i}{x_i} \right| \times 100 \quad (8)$$

同时为了更好的展示模型对于趋势的预测准确率，也采用趋势准确率作为评价指标。

$$\text{趋势准确率} = \frac{\text{趋势预测正确的数量}}{\text{总个数}} \quad (9)$$

4. 风险控制

4.1 主要风险

风险主要分为系统性风险（投资者很难避免）和非系统性风险，非系统性风险主要分为市场风险和模型预测风险：

市场风险：包括价格波动风险、流动性风险、系统性风险、利率风险和信用风险。价格波动风险源于经济数据、政策变化和 international 事件等多种因素，导致市场价格剧烈波动并影响投资回报。流动性风险在市场流动性不足时，可能导致难以及时以合理价格买入或卖出资产，增加交易成本或无法及时止损。系统性风险在市场整体下跌时，即使模型预测准确，投资组合也可能受到重大影响，如经济衰退和金融危机等。利率风险因利率变动影响市场资金成本，进而影响资产价格，例如利率上升可能导致股票和债券价格下跌。信用风险涉及债券等固定收益资产时，发行主体可能因财务困境无法按时偿还本金和利息，导致投资损失。

模型预测风险：包括模型不准确性风险、数据质量风险、过拟合风险、模型滞后风险、参数选择风险和技术风险。即使经过优化和再训练，模型在实际市场中的预测仍可能不准确，导致投资决策失误。模型的预测依赖于输入数据的质量，如果数据存在噪声、不完整或不准确，会影响模型的预测性能。过拟合风险指模型在训练集上表现良好，但在未见过的数据上表现不佳，影响实际投资效果。市场环境变化迅速，模型可能无法及时适应新的市场状况，导致预测滞后。模型的性能依赖于参数选择，参数选择不当会导致模型预测性能下降。技术风险包括计算机系统故障、软件错误、数据存储损坏等技术问题，可能影响模型的正常运行和预测结果。

为了尽量减少市场风险，我们设置止损线，根据 ATR 指标设置止损线，及时止损，控制单笔交易的最大亏损。

为了尽量减少模型预测风险，我们构造了模型的止损策略，如果模型出现多次失误，将暂停使用该模型，并重新进行适量训练以寻找最优参数。

4.2 ATR 止损策略

我们构造了 ATR 止损指标，ATR（平均真实波动范围）是一种常用的衡量市场波动性的指标。ATR 的计算方法是取一定周期内价格的最高价、最低价和收盘价之间波动的平均值。具体计算公式为：

$$ATR = \frac{\text{前一周期的} ATR \times (N-1) + \text{本周期的} TR}{N}$$

其中，N 为周期数，TR（真实范围）为以下三者中的最大值：

1. 本周期最高价和最低价的差
2. 前一周期的收盘价与本周期最高价的差
3. 前一周期的收盘价与本周期最低价的差

止损线的设置为：

$$\text{止损线} = \text{进场价格} \pm (\text{ATR} \times \text{止损系数})$$

其中，止损系数是根据风险偏好设定的参数。当股票价格触及以 ATR 为基础的止损线时，即触发止损操作。

4.3 模型止损策略

同时我们构造了模型的止损策略，如果模型出现多次失误，将暂停使用该模型，并重新进行适量训练以寻找最优参数。具体来说，当模型的连续失误次数达到设定的阈值，或在一段时间内的总体预测准确率低于设定的最低值（如 50%），则触发止损策略，暂停模型使用并进行再训练。当模型的预测准确率提升至设定的阈值后（如 75%），将重新启用该模型进行投资。

假设当前处于第 i 个时间步（ $i > 2$ ），我们首先计算历史股票趋势：

$$\text{true_trend} = \text{stock_price}_{i-1} - \text{stock_price}_{i-2} \quad (10)$$

然后计算预测股票趋势：

$$\text{predicted_trend} = \text{predict_price}_{i-1} - \text{stock_price}_{i-2} \quad (11)$$

如果预测股票趋势与历史股票趋势同号（即 $\text{predicted_trend} \times \text{true_trend} > 0$ ），则在计数器 counter 中记录+1；否则记录-1。如果 counter 连续出现 5 次-1，则停止任何投资策略（模型仍在预测，但取消任何买入或抛出），并进行参数优化。直到 counter 连续出现三次+1，重新采用模型。

5 迁移学习及参数冻结

本节将讨论模型随着步长的累加导致的误差，以及如何运用迁移学习解决带来的问题。

5.1 预测误差问题

随着时间的推移，模型的性能会出现下降的变化。

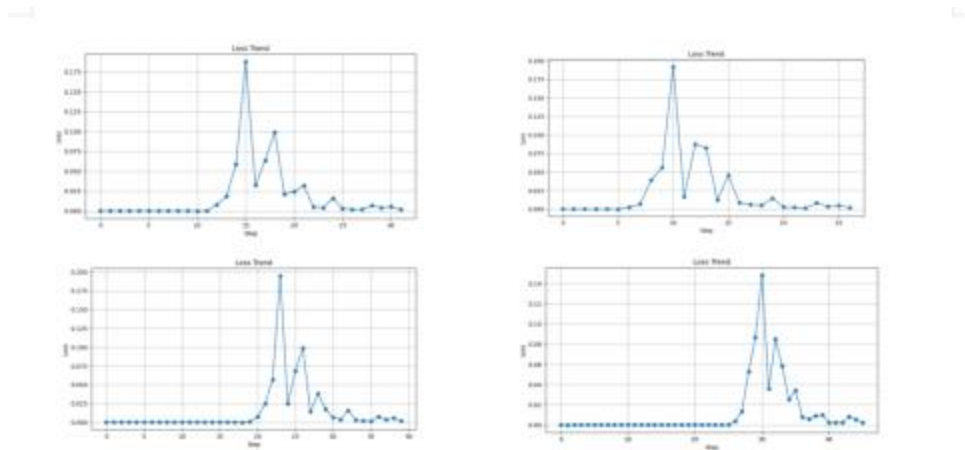


图 5.1 损失函数趋势图

图 5.1 是截取的不同股票在某段时间内损失函数的趋势图。可以看到，在一定的预测周期后，Informer 模型出现了 Loss 的突然上升，即不可避免的“性能衰减”。这是因为深度学习模型从历史数据中学习经验，而当面对越来越多的未知数据时，模型的预测误差不可避免地会增加。

因此，需要提高模型的训练频率，而不是期望一次训练即可长期维持高性能。然而，完全重新训练模型在新数据量不大的情况下容易导致过拟合，并且训练过程过于耗时。为了解决这个问题，团队引入了迁移学习中的“参数冻结”方法。

通过“参数冻结”方法，团队只对模型的一部分参数进行更新，而其他参数保持不变。这种方式既能有效利用新数据提高模型性能，又能避免过拟合和减少训练时间。通过这种综合策略，在不显著增加计算成本的前提下，确保模型长时间保持高水平的预测能力。

5.2 参数冻结

参数冻结是一种训练神经网络模型时的技术，它指的是在模型的训练过程中固定（或“冻结”）一部分模型参数，不对其进行更新。通常情况下，参数冻结应用于预训练模型的微调过程中，特别是大型的预训练模型，想要将其应用于新数据集时。在参数冻结中，一般会将模型的一部分或全部层的参数固定，然后只

更新模型的部分参数。这样做的目的是保留模型在预训练任务上学到的特征表示，同时允许模型适应新的任务或数据集。通过冻结部分参数，可以减少模型在新任务上的训练参数数量，从而减少了过拟合的风险，并提高了模型的泛化能力。总的来说，参数冻结是一种有效的策略，可以帮助在新任务上更好地利用预训练模型的知识，加快模型的收敛速度，并提高模型的性能。

团队使用参数冻结方法，对除去最后两层全连接层外的所有可训练参数设置为不可梯度下降，从而大幅度提高训练时间（以一个 epoch 为计算时间），一个模型训练 1000 个 epoch 所消耗的时间约为 31 小时（3090 显卡），而使用参数冻结后所消耗的时间约为 3 个半小时，时间提升了近 10 倍。

表 5.1 参数冻结与参数未冻结时间对比实验

股票	未冻结参数的时间	冻结参数后的时间
000001. SZ	1min52s	15.642s
000001. SZ	1min54s	17.1s
000001. SZ	1min48s	11.93s

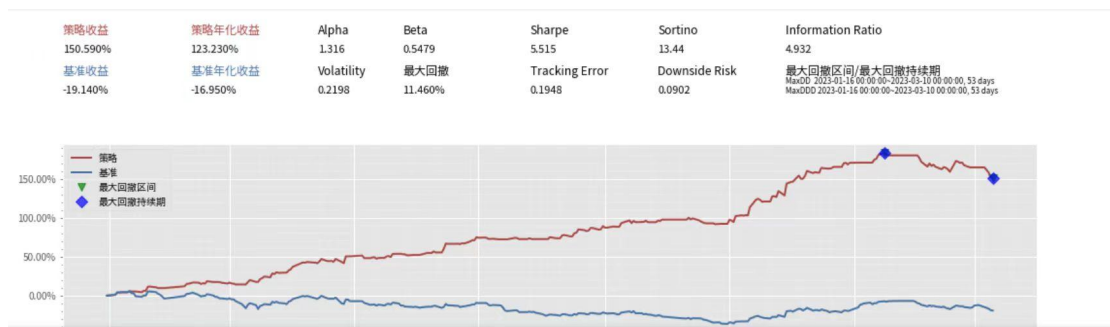
表 5.2 参数冻结与参数未冻结损失函数对比实验

股票	未冻结参数的 Loss	冻结参数后的 Loss
000001. SZ	0.001184	0.001131
000001. SZ	0.001362	0.001364
000001. SZ	0.001407	0.001389

可以明显看到时间即为原先的 1/3，即进行一个未冻结参数训练的时间相当于 3 次冻结参数训练的时间。同时冻结参数后的 Loss 总的上来看也是略小于未冻结参数的 Loss。究其根本是因为新数据不符合原先股票数据的特征分布，导致模型无法充分学习，此时如果不冻结参数，会导致模型过拟合问题，导致预测能力下降。

6 回测

6.1 2023 年回测



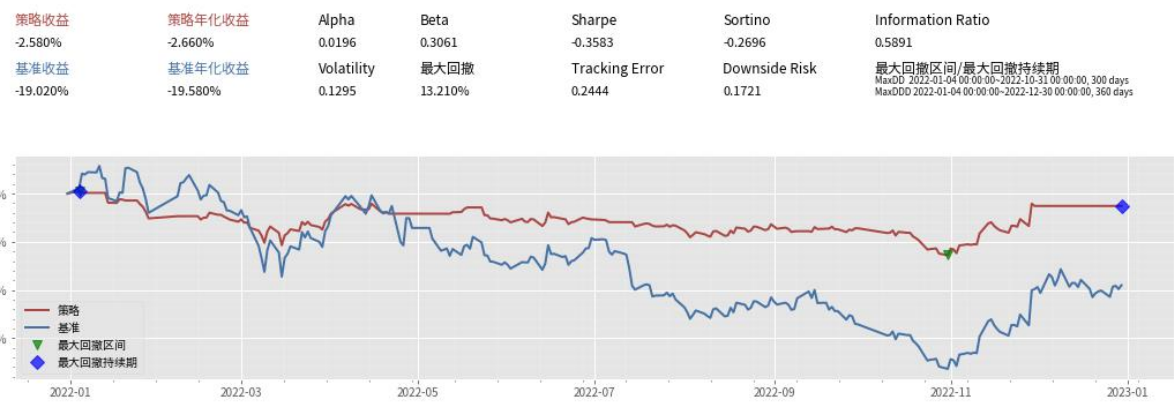
回测时间段：2023 年 1 月 1 日—2023 年 6 月 1 日

资金：10000000

策略收益：150.590%

基准收益：-19.140%

6.2 2022 年回测



回测时间段：2022 年 1 月 1 日—2023 年 12 月 31 日

资金：10000000

策略收益：-7.540%

基准收益：-19.020%

6.3 2024 年回测



回测时间段：2024 年 1 月 1 日—2024 年 6 月 1 日

资金：10000000

策略收益：16.960%

基准收益：18.320%

7 总结

团队构建了面向股票趋势预测的 Informer 模型，旨在预测未来股票走势，从而更好的做出决策。不同于其他面向收益率的传统止损策略，团队采取面向模型的止损策略，赋予模型更大的决策权。回测结果表明，Informer 具有良好的预测趋势能力。同时团队通过冻结参数方法，使用每日新增的数据对模型结构做出修正，从而提高预测精度。最后，团队还发现在分钟线上，Informer 具有更加强大的预测能力，这得益于时间戳处理机制，但是由于数据量过大，算力不足，并未有时间进行全面测试。在未来，团队将填补分钟线交易策略的空白，继续挖掘深度学习模型与量化投资的关系。

8 参考文献

- [1] Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *Journal of Finance*, 19(3), 425–442.
- [2] Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques - Part II: Soft computing methods. *Expert Systems with Applications*, 36(3), 5932–5941.
- [3] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- [5] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12), 11106–11115.
- [6] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(12), 809

