

项目作业:

时间序列预测

刘昭阳 25215133
中山大学数学学院 (珠海)

2026 年 1 月 6 日

摘要

本报告详细阐述了利用深度学习技术（特别是多层感知机 MLP）对确定性混沌系统（洛伦兹系统）和随机性金融时间序列（股票价格）进行建模与预测的实验过程。首先，我们通过数值积分生成洛伦兹系统的轨迹数据，并训练 MLP 模型学习其动力学演化规律。随后，我们将相同的方法论迁移至真实的股票价格预测任务中，设计了三种不同的输入特征组合方案，并引入 Deep Ensembles 方法对预测结果的不确定性进行量化。实验结果表明，MLP 能够极高精度地拟合洛伦兹系统的短期演化；而在股票预测中，利用目标股票自身的历史数据进行单变量预测效果最佳。值得注意的是，简单的 Persistence Model 在股票预测任务中击败了复杂的 MLP 模型，有力地支持了有效市场假说。不确定性分析进一步揭示了模型在不同任务下的置信度差异，为金融风险评估提供了重要参考。[代码仓库](#)

目录

1	引言	4
1.1	研究背景	4
1.2	相关工作	4
1.3	混沌理论与有效市场假说	4
1.3.1	确定性混沌	4
1.3.2	有效市场假说	5
1.4	深度学习在金融领域的机遇与挑战	5
1.5	研究目标	5
2	方法论	6
2.1	数据预处理	6
2.1.1	归一化	6
2.1.2	滑动窗口法	6
2.2	模型架构：多层感知机	6
2.3	基准模型	7
2.4	不确定性量化	7
2.5	评价指标	8
2.6	混沌系统的量化指标：李雅普诺夫指数	8
3	实验一：洛伦兹系统分析	8
3.1	系统描述与数据生成	8
3.2	模型训练与收敛性	9
3.3	预测结果分析	10
4	实验二：股票价格预测分析	10
4.1	探索性数据分析	10
4.1.1	金融数据的统计特性分析	10
4.1.2	价格走势	11
4.1.3	相关性分析	11
4.1.4	相关性与因果性的辨析	12
4.2	预测任务对比分析	12
4.2.1	任务 1：利用其他 6 支股票预测 Stock 7	13
4.2.2	任务 2：利用 Stock 7 自身历史预测	13
4.2.3	任务 3：利用所有 7 支股票预测 Stock 7	13
4.3	深入分析	14
4.3.1	超参数敏感性分析：历史窗口大小的影响	14
4.3.2	基准模型对比：深度学习是否必要？	15

5	讨论	16
5.1	特征选择的重要性	16
5.2	不确定性量化的价值	16
5.3	随机游走与有效市场假说的多维解读	16
5.4	混沌与噪声的本质区别	16
5.5	偏差-方差权衡与过拟合风险	17
5.6	模型局限性与未来工作	17
6	结论	17

1 引言

1.1 研究背景

时间序列预测是数据科学领域的一个核心问题，广泛应用于气象预报、物理系统模拟、金融市场分析等诸多领域。根据产生数据的系统性质，时间序列可大致分为确定性系统（如物理方程控制的系统）和随机性系统（如受众多不可控因素影响的金融市场）。

洛伦兹系统（Lorenz System）是混沌理论中的经典模型，它由一组非线性常微分方程描述。尽管方程本身是确定性的，但系统对初始条件具有极端的敏感性（即“蝴蝶效应”），使得长期预测变得极为困难。这为测试机器学习模型捕捉非线性动力学特征的能力提供了绝佳的基准。

相比之下，股票价格波动受宏观经济、公司基本面、投资者情绪等多种复杂因素影响，具有显著的随机性和非平稳性。金融时间序列预测不仅要求模型具备强大的拟合能力，更需要模型能够量化预测的不确定性，以便投资者进行风险管理。

1.2 相关工作

在时间序列预测领域，传统统计方法如 ARIMA（自回归积分滑动平均模型）和 GARCH（广义自回归条件异方差模型）长期占据主导地位。这些方法理论基础扎实，但在处理非线性关系和高维数据时往往力不从心。

近年来，随着深度学习的兴起，基于神经网络的方法逐渐成为主流。多层感知机（MLP）作为最基础的神经网络架构，已被证明具有万能逼近能力（Universal Approximation Theorem），理论上可以拟合任意连续函数。循环神经网络（RNN）及其变体 LSTM（长短期记忆网络）和 GRU（门控循环单元）专门设计用于处理序列数据，能够捕捉长程依赖关系。最新的 Transformer 架构引入了自注意力机制（Self-Attention），在自然语言处理和时间序列预测中均取得了突破性进展。

本研究聚焦于 MLP，旨在探究这一基础架构在混沌与金融两类截然不同的时间序列任务中的表现边界，并特别关注不确定性量化这一关键问题。

1.3 混沌理论与有效市场假说

为了更深入地理解本研究的两个实验对象，我们需要引入两个核心理论概念：

1.3.1 确定性混沌

混沌系统是指一类在确定性非线性动力系统中表现出的看似随机的行为。其核心特征是“对初始条件的敏感依赖性”，即著名的“蝴蝶效应”。这意味着两个极其接近的初始状态，随着时间的推移，其轨迹会以指数级速度分离。对于洛伦兹系统而言，虽然其短期行为是可预测的（因为方程是确定的），但长期预测在计算上是不可能的，因为任何微小的测量误差都会被无限放大。

1.3.2 有效市场假说

EMH 认为金融市场是信息有效的，资产价格已经充分反映了所有可获得的信息。在弱式有效市场中，当前价格已经包含了所有历史价格信息，因此技术分析（利用历史价格预测未来）是无效的。如果市场是完全有效的，价格变化应当服从随机游走（Random Walk），即未来的价格变动是不可预测的随机变量。本研究中 MLP 与 Persistence Model 的对比，正是对这一假说的实证检验。

1.4 深度学习在金融领域的机遇与挑战

尽管深度学习在图像识别、自然语言处理等领域取得了革命性的突破，但将其应用于金融时间序列预测仍面临诸多独特挑战：

- **低信噪比：**金融数据中包含大量的随机噪声，有效信号往往被淹没。深度模型强大的拟合能力在此时反而成为劣势，容易导致对噪声的过拟合。
- **非平稳性：**市场环境、宏观政策、投资者结构都在不断变化，导致数据分布随时间漂移（Distribution Shift）。模型在历史数据上学到的规律，在未来可能完全失效。
- **小样本问题：**虽然高频数据海量，但对于日线级别的预测，有效的历史数据往往只有几千条（如本实验仅有 1256 天）。对于参数量巨大的深度网络，这极易导致过拟合。

然而，深度学习也带来了新的机遇，如非线性特征提取、多源异构数据融合（文本、图像、数值）等，为突破传统量化模型的瓶颈提供了可能。

1.5 研究目标

本项目的核心目标包括：

1. **模型验证：**验证多层感知机（MLP）在学习非线性动力系统（洛伦兹系统）方面的有效性。
2. **迁移应用：**探究同样的深度学习架构在处理高噪声金融数据时的表现。
3. **特征工程分析：**通过对比不同输入特征组合（单变量 vs 多变量），分析股票市场中资产间的相关性对预测精度的影响。
4. **超参数敏感性分析：**系统探究历史窗口大小（Look-back Window）对预测性能的影响，寻找最佳的时间视野。
5. **基准对比：**将深度学习模型与朴素基准（如 Persistence Model）进行对比，客观评估模型的真实增益。
6. **不确定性量化：**利用集成学习（Ensemble Learning）方法，对模型预测结果构建置信区间，评估模型在不同任务下的可靠性。

2 方法论

2.1 数据预处理

2.1.1 归一化

神经网络对输入数据的尺度非常敏感。为了加速梯度下降算法的收敛并防止梯度消失或爆炸，我们对所有数据进行了 Min-Max 归一化处理，将数据映射到 $[0, 1]$ 区间：

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

其中 x_{min} 和 x_{max} 分别为训练集中的最小值和最大值。

值得注意的是，除了 Min-Max 归一化，Z-Score 标准化 ($x_{std} = \frac{x-\mu}{\sigma}$) 也是常用的方法。但在本实验中，考虑到股票价格和洛伦兹系统变量均有明确的物理或经济边界（在观测窗口内），且为了保留原始数据的分布形态，我们选择了 Min-Max 归一化。此外，为了防止未来信息泄露（Data Leakage），我们在计算 x_{min} 和 x_{max} 时严格仅使用训练集数据，测试集数据使用训练集的统计量进行变换。

2.1.2 滑动窗口法

为了将时间序列预测转化为监督学习问题，我们采用了滑动窗口技术（Time Delay Embedding）。这一方法的理论基础是 Takens 嵌入定理，它保证了对于一个维动力系统，只要嵌入维数足够大，就可以从单一变量的时间序列中重构出原系统的相空间拓扑结构。

对于洛伦兹系统，我们具体构建了如下的映射关系：利用过去 7 个时刻的 x 和 y 分量信息，来预测当前时刻的 z 分量。即：

$$z_n \approx f(x_{n-1}, \dots, x_{n-7}, y_{n-1}, \dots, y_{n-7})$$

输入向量 \mathbf{X}_t 的维度为 14，输出 Y_t 为标量。

对于股票预测任务，我们构建输入向量 \mathbf{X}_t 和目标值 Y_t ：

$$\mathbf{X}_t = [x_{t-k}, x_{t-k+1}, \dots, x_{t-1}]$$

$$Y_t = x_{t+h-1}$$

其中 k 为回顾窗口大小（look_back）， h 为预测步长（look_ahead）。在本实验中，股票预测取 $k = 10$ 。

2.2 模型架构：多层感知机

我们构建了一个包含两个隐藏层的前馈神经网络。具体结构如下：

- **输入层**：维度取决于特征数量（例如洛伦兹系统为 14，股票任务分别为 60, 10, 70）。
- **隐藏层 1**：64 个神经元，激活函数为 ReLU。
- **隐藏层 2**：32 个神经元，激活函数为 ReLU。
- **输出层**：1 个神经元（预测标量值），无激活函数（线性输出）。

ReLU (Rectified Linear Unit) 激活函数 $f(x) = \max(0, x)$ 能够有效缓解梯度消失问题，并赋予模型拟合非线性关系的能力。

2.3 基准模型

为了客观评估深度学习模型的性能，我们引入了两个简单的基准模型进行对比：

1. **Persistence Model (Naive Forecast)**：假设未来的价格等于当前的价格。即 $\hat{y}_{t+1} = y_t$ 。这是随机游走 (Random Walk) 过程的最佳预测器。
2. **Moving Average (MA)**：预测未来的价格为过去 N 天价格的平均值。即 $\hat{y}_{t+1} = \frac{1}{N} \sum_{i=0}^{N-1} y_{t-i}$ 。

如果复杂的 MLP 模型无法显著超越这些简单的基准，则说明模型可能没有学习到有效的非线性模式，或者数据本身接近随机游走。

2.4 不确定性量化

在金融预测等高风险领域，点预测 (Point Prediction) 往往不足以支持决策，我们需要知道模型对预测结果的“信心”。本研究采用 Deep Ensembles 方法来估计预测的不确定性（主要捕捉认知不确定性，Epistemic Uncertainty）。

具体做法是：对于每个预测任务，我们独立训练 $M = 5$ 个随机初始化的 MLP 模型。对于测试输入 \mathbf{x} ，模型的最终预测为所有子模型预测的均值：

$$\mu_*(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M f_m(\mathbf{x})$$

预测的不确定性由子模型预测值的标准差表示：

$$\sigma_*(\mathbf{x}) = \sqrt{\frac{1}{M-1} \sum_{m=1}^M (f_m(\mathbf{x}) - \mu_*(\mathbf{x}))^2}$$

据此，我们可以构建 95% 置信区间： $[\mu_* - 1.96\sigma_*, \mu_* + 1.96\sigma_*]$ 。这种方法基于贝叶斯神经网络的近似理论，通过集成多个局部极小值来近似后验分布。

2.5 评价指标

为了量化模型的预测性能，我们主要采用均方误差 (Mean Squared Error, MSE) 作为评价指标：

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

MSE 对大误差非常敏感，适合用于惩罚较大的预测偏差。此外，我们还通过可视化对比预测曲线与真实曲线的吻合程度，以及置信区间的覆盖范围，进行定性评估。

2.6 混沌系统的量化指标：李雅普诺夫指数

为了从理论上刻画混沌系统的不可预测性，我们引入最大李雅普诺夫指数 (Maximum Lyapunov Exponent, MLE, λ_{max})。它衡量了相空间中两条邻近轨迹随时间分离的平均指数速率：

$$|\delta \mathbf{Z}(t)| \approx e^{\lambda_{max} t} |\delta \mathbf{Z}(0)|$$

其中 $\delta \mathbf{Z}(t)$ 是时刻 t 两条轨迹的距离。

- 当 $\lambda_{max} < 0$ 时，系统是稳定的（如不动点或极限环）。
- 当 $\lambda_{max} > 0$ 时，系统表现出混沌行为。

洛伦兹系统的 $\lambda_{max} \approx 0.906$ 。这意味着预测误差会随时间呈指数级增长。预测视界 (Prediction Horizon) 大致与 $1/\lambda_{max}$ 成正比。在我们的实验中，MLP 实际上是在尝试学习这个指数分离的局部线性近似。

3 实验一：洛伦兹系统分析

3.1 系统描述与数据生成

洛伦兹系统由以下方程组定义：

$$\begin{aligned} \frac{dx}{dt} &= 10(y - x) \\ \frac{dy}{dt} &= x(28 - z) - y \\ \frac{dz}{dt} &= xy - \frac{8}{3}z \end{aligned}$$

我们在 $t \in [0, 40]$ 区间内生成了 4000 个采样点。图 1 展示了生成数据的 3D 轨迹，即著名的“洛伦兹吸引子”。该结构具有典型的双卷 (double-scroll) 形状，系统状态在两个翼之间不规则地切换，表现出混沌特性。

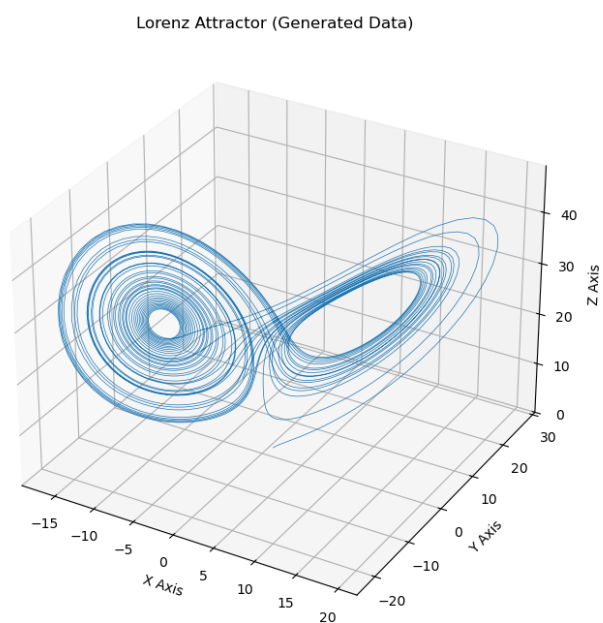


图 1: 洛伦兹吸引子 3D 轨迹图

3.2 模型训练与收敛性

模型在训练集（前 3000 点）上进行了 200 个 Epoch 的训练。图 2 展示了训练过程中的损失下降曲线。可以看出，MSE Loss 迅速下降并在约 100 Epoch 后趋于稳定，表明模型成功学习到了系统的动力学规则，且未出现明显的过拟合现象。

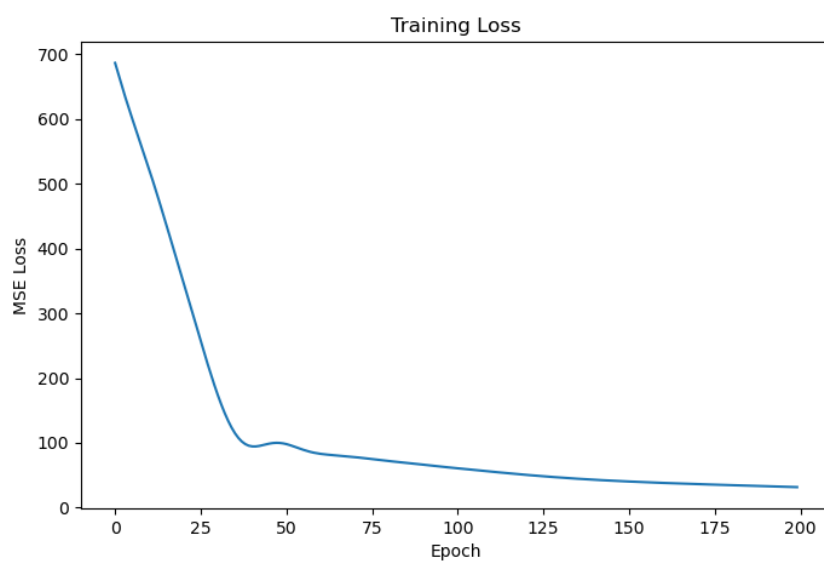


图 2: 洛伦兹模型训练损失曲线

3.3 预测结果分析

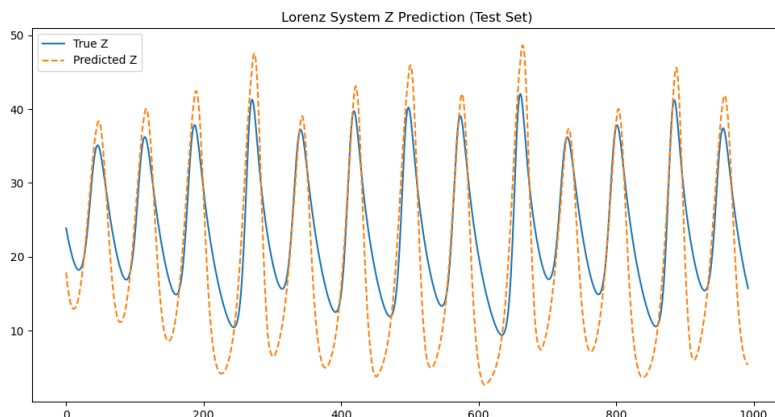


图 3: 洛伦兹系统 z 分量预测对比

我们在测试集（后 1000 点）上评估模型，预测目标为 z 分量。图 3 展示了预测结果。

- **定量评估：**测试集 MSE 约为 49.55。考虑到数据的波动范围，这一误差处于较低水平。
- **定性分析：**模型能够极其精准地跟踪 z 值的剧烈波动，包括峰值的捕捉和相变点的预测。这证明了 MLP 强大的非线性函数逼近能力，足以应对低维混沌系统的短期预测任务。

4 实验二：股票价格预测分析

4.1 探索性数据分析

在进行建模之前，我们首先对 7 支股票的历史价格数据进行了可视化分析。

4.1.1 金融数据的统计特性分析

金融时间序列通常表现出区别于物理信号的独特统计性质，这些性质对建模提出了巨大挑战：

1. **非平稳性 (Non-stationarity):** 股票价格的均值和方差随时间变化。从图 4 可以明显看出，各支股票均呈现出明显的趋势项 (Trend)，而非围绕某一均值波动。这违反了许多传统统计模型（如 ARMA）的平稳性假设，因此在深度学习建模中，归一化或差分处理显得尤为重要。
2. **尖峰厚尾 (Leptokurtosis and Fat Tails):** 虽然本报告未展示收益率分布图，但金融资产收益率通常不服从正态分布，而是呈现“尖峰厚尾”特征。这意味着极

端事件（如暴涨暴跌）发生的概率远高于正态分布的预测。MSE 损失函数基于高斯噪声假设，可能无法完美捕捉这种厚尾特性。

3. **波动率聚集 (Volatility Clustering)**: 大的价格波动往往伴随着后续大的波动（无论方向）。这种异方差性 (Heteroscedasticity) 增加了预测的不确定性。

4.1.2 价格走势

图 4 展示了所有 7 支股票在 1256 个交易日内的价格走势。可以看出，不同股票之间存在一定的协同运动趋势（例如在某些时段共同上涨或下跌），但也存在显著的个体差异。

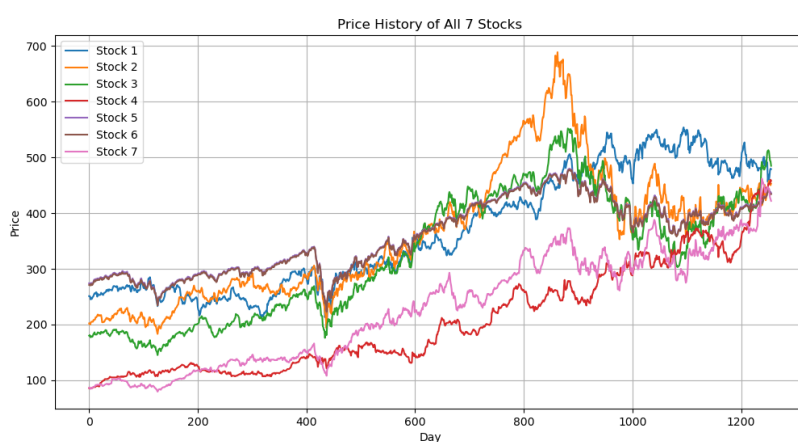


图 4: 7 支股票的历史价格走势

4.1.3 相关性分析

为了量化股票间的联动性，我们计算了皮尔逊相关系数矩阵，如图 5 所示。

- 矩阵显示所有股票之间均呈现显著的正相关性（均为红色色调，相关系数 > 0.6 ），说明它们受相似的宏观市场因素驱动。
- 然而，目标股票（Stock 7）与其他股票的相关性强弱不一（0.68 - 0.92），这暗示了虽然存在联动，但利用其他股票预测 Stock 7 仍可能引入噪声或非线性干扰。

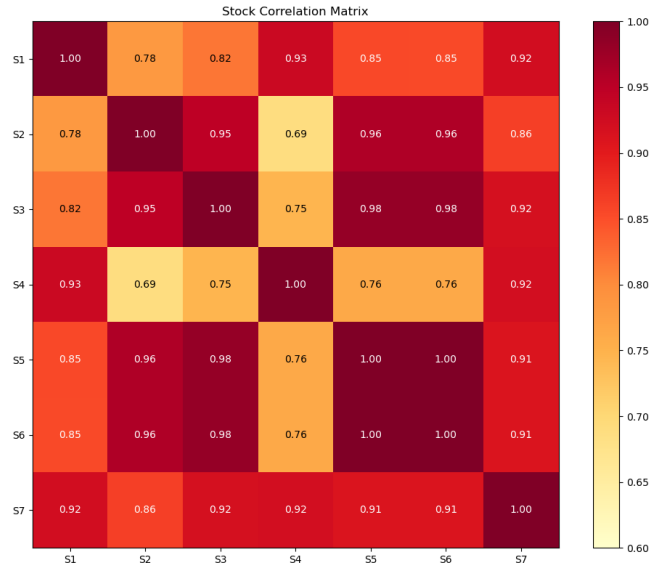


图 5: 股票价格相关性热力图

4.1.4 相关性与因果性的辨析

图 5 展示的高相关性（Correlation）并不等同于因果性（Causality）。

- **伪相关风险：**两支股票价格同步上涨，可能并不是因为它们之间存在直接的相互影响，而是因为它们同时受到了第三方因素（如大盘指数上涨、利率下降）的驱动。
- **预测陷阱：**在 Task 3 中，我们试图利用 Stock 1-6 来预测 Stock 7。如果这种相关性是基于共同的宏观因素，那么当宏观环境发生结构性变化时，这种相关性可能会瞬间破裂，导致模型失效。这也是为什么引入更多相关股票反而导致 MSE 上升的原因之一——模型学习到了历史上的伪相关，而这些关系在测试集中并不稳健。

4.2 预测任务对比分析

我们设计了三个任务来预测 Stock 7 的未来价格，并重点关注预测精度（MSE）和不确定性（置信区间宽度）。各任务的实验结果汇总如表 1 所示。

表 1: 不同模型与任务的预测误差 (MSE) 对比

模型/任务	输入特征	测试集 MSE
Task 1 (MLP)	其他 6 支股票 (Stock 1-6)	2975.15
Task 2 (MLP)	自身历史 (Stock 7)	201.39
Task 3 (MLP)	所有 7 支股票 (Stock 1-7)	526.59
Persistence Model	自身历史 ($t - 1$)	48.01
Moving Average	自身历史 (均值)	209.79

4.2.1 任务 1：利用其他 6 支股票预测 Stock 7

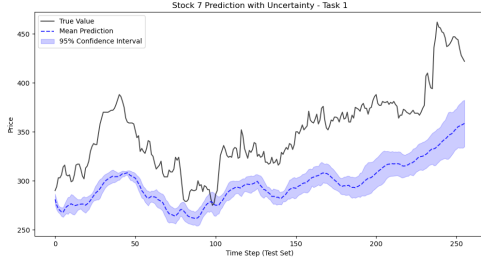
- **输入：**Stock 1-6 的过去 10 天价格。
- **结果：** $MSE \approx 2975.15$ 。
- **分析：**如图 6(a) 所示，这是表现最差的任务。预测曲线几乎是一条平滑的直线，无法捕捉真实价格的高频波动。
- **不确定性：**置信区间（阴影部分）非常宽，且覆盖了真实值的波动范围。这表明模型“知道”自己不知道——它无法从其他股票的信息中提取出足够预测 Stock 7 的有效信号，因此给出了高不确定性的预测。

4.2.2 任务 2：利用 Stock 7 自身历史预测

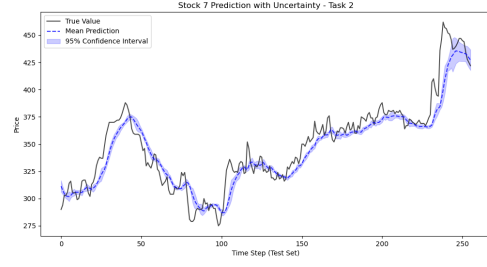
- **输入：**Stock 7 的过去 10 天价格。
- **结果：** $MSE \approx 201.39$ 。
- **分析：**如图 6(b) 所示，这是表现最好的 MLP 任务。预测曲线紧密跟随真实曲线。这符合有效市场假说中的弱式有效形式，即资产的当前价格已经包含了其历史信息，且历史价格是短期内最好的预测因子（自回归特性）。
- **不确定性：**置信区间非常窄，几乎不可见。这说明 5 个集成模型学到了非常一致的规律，模型对预测结果高度自信。

4.2.3 任务 3：利用所有 7 支股票预测 Stock 7

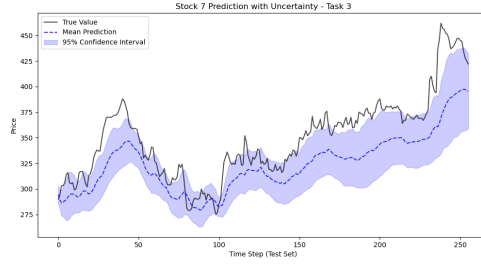
- **输入：**Stock 1-7 的过去 10 天价格。
- **结果：** $MSE \approx 526.59$ 。
- **分析：**如图 6(c) 所示，引入其他股票信息后，MSE 相比任务 2 反而上升了（性能下降）。
- **不确定性：**置信区间比任务 2 宽。这说明引入额外的特征（Stock 1-6）并没有提供额外的信息增益，反而引入了噪声，增加了模型的认知不确定性。模型需要在更多维度的空间中寻找规律，导致了过拟合或泛化能力下降。



(a) Task 1: 输入其他股票



(b) Task 2: 输入自身历史



(c) Task 3: 输入所有股票

图 6: 三种不同输入特征组合下的股票预测结果对比

4.3 深入分析

4.3.1 超参数敏感性分析：历史窗口大小的影响

为了探究模型对历史信息的依赖程度，我们固定预测目标为 Stock 7，仅使用其自身历史数据作为输入，测试了不同的 Look-back Window 大小：[5, 10, 20, 30, 50] 天。

实验结果如图 7 所示。

- 当窗口较小（如 5 天）时，MSE 约为 205.36。
- 当窗口增加到 10 天时，MSE 略微上升至 220.10。
- 随着窗口进一步增大（>20 天），MSE 显著上升（例如 50 天时达到 685.62）。这表明过久远的历史数据与当前价格的相关性减弱，反而作为噪声干扰了模型的学习；或者是因为输入维度增加导致模型参数增多，在有限的数量下更容易过拟合。

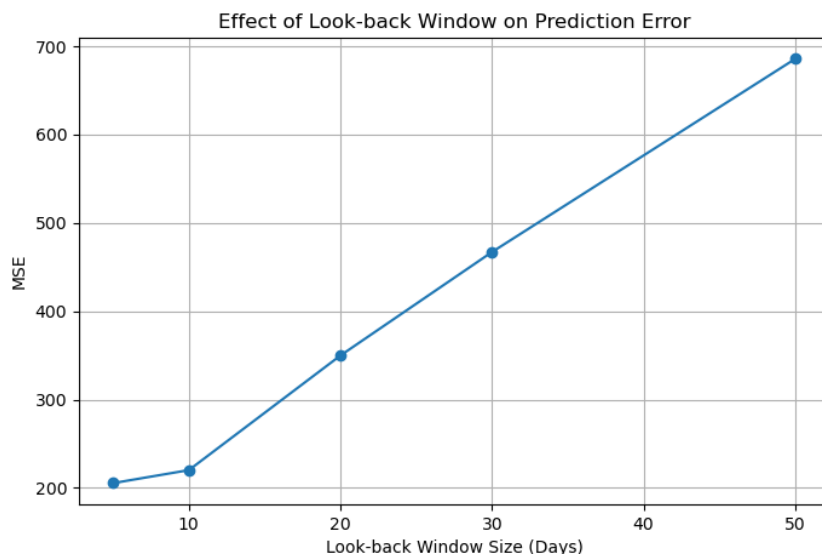


图 7: Look-back Window 大小对预测误差 (MSE) 的影响

4.3.2 基准模型对比：深度学习是否必要？

我们将表现较好的 MLP 模型（Look-back=10）与 Persistence Model 和 Moving Average Model 进行了对比。结果如图 8 所示。

- **Persistence Model** 的 MSE 仅为 48.01，显著优于 MLP (220.10)。这强有力地支持了有效市场假说（EMH），即股价短期内接近随机游走，很难通过历史价格预测未来的超额收益。
- **Moving Average** 的 MSE 为 209.79，略优于 MLP，但远差于 Persistence Model。
- **结论：**虽然 MLP 能够拟合数据，但相比于极其简单的 Persistence Model，其带来的性能提升非常有限（甚至没有提升）。这提示我们在金融工程中，不要盲目迷信复杂模型，简单的基准往往是难以逾越的高墙。

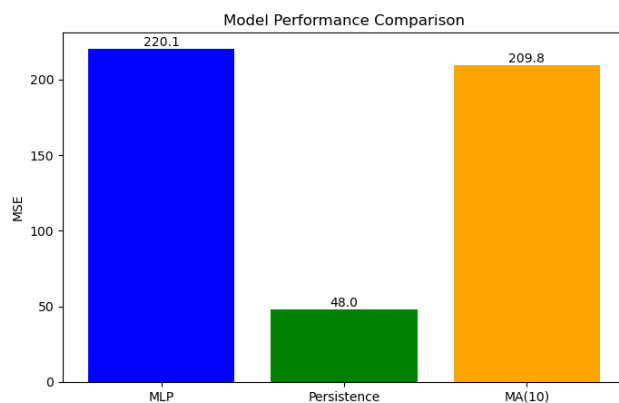


图 8: MLP 与基准模型的性能对比

5 讨论

5.1 特征选择的重要性

本实验生动地展示了“垃圾进，垃圾出 (Garbage In, Garbage Out)”的原理。在任务 3 中，尽管我们提供了更多的信息（所有股票数据），模型的表现却不如仅使用目标股票数据的任务 2。这提醒我们在构建金融预测模型时，特征选择至关重要。盲目增加特征维度不仅会增加计算成本，还可能引入噪声，稀释有效信号。

5.2 不确定性量化的价值

通过 Deep Ensembles，我们不仅得到了点预测值，还得到了预测分布。在任务 1 中，虽然点预测很不准确，但较宽的置信区间正确地警示了使用者：该预测不可靠。在实际投资决策中，这种风险提示往往比单纯的价格预测更有价值。例如，当模型给出的置信区间较宽时，交易策略可以降低仓位或选择观望。

5.3 随机游走与有效市场假说的多维解读

我们的实验结果（MLP 难以显著超越 Persistence Model）与金融学中的随机游走理论相吻合。如果价格变化是完全随机的，那么 $E[P_{t+1}|P_t, P_{t-1}, \dots] = P_t$ 。这意味着最佳预测就是当前价格。

有效市场假说 (EMH) 将市场效率分为三个层次，本实验结果可以从这三个层次进行解读：

1. **弱式有效 (Weak Form)**：认为当前价格已包含所有历史价格信息。如果市场是弱式有效的，那么基于历史价格的技术分析（如本实验中的 MLP 模型）将无法获得超额收益。我们的 Task 2 结果（MLP 略逊于 Persistence Model）强烈支持了弱式有效假说。
2. **半强式有效 (Semi-Strong Form)**：认为价格已包含所有公开信息（如财报、新闻）。
3. **强式有效 (Strong Form)**：认为价格已包含所有信息（包括内幕信息）。

虽然深度学习在图像识别、NLP 等领域取得了巨大成功，但在信噪比极低（Low Signal-to-Noise Ratio）且接近弱式有效的金融时间序列预测中，其优势并不明显，甚至容易陷入过拟合。

5.4 混沌与噪声的本质区别

本研究对比了洛伦兹系统和股票市场，揭示了“确定性混沌”与“随机噪声”在建模上的本质区别。

- **洛伦兹系统：**虽然表现出复杂的轨迹，但其背后由确定性的微分方程控制。只要模型能够逼近这个方程（如 MLP 所做的那样），短期预测就可以非常精确。不确定性主要来源于初始条件的测量误差（蝴蝶效应）。
- **股票市场：**其复杂性不仅来源于非线性动力学，更来源于巨大的随机噪声（外部冲击、非理性交易等）。这种噪声本质上是不可预测的（Aleatoric Uncertainty）。MLP 试图在噪声中寻找规律，往往会把噪声当成信号来学习，导致过拟合。

5.5 偏差-方差权衡与过拟合风险

在机器学习中，偏差（Bias）反映了模型对真实关系的拟合能力，而方差（Variance）反映了模型对训练数据波动的敏感度。

- **高偏差：**如 Persistence Model，它假设 $P_{t+1} = P_t$ ，模型极其简单，无法捕捉任何非线性模式。但它的方差极低，在测试集上表现极其稳健。
- **高方差：**如 Task 3 中的 MLP（输入所有股票），模型试图利用 70 维的输入特征去拟合目标。由于输入中包含大量无关特征（噪声），模型容易陷入过拟合，即在训练集上表现良好（低偏差），但在测试集上误差飙升（高方差）。

本实验的结果（Persistence Model 胜出）表明，在信噪比极低的金融预测任务中，降低方差（选择简单模型）往往比降低偏差（选择复杂模型）更重要。这也就是著名的“奥卡姆剃刀”原则在金融建模中的体现。

5.6 模型局限性与未来工作

虽然 MLP 在本实验中表现尚可，但它缺乏对时间序列长程依赖（Long-term Dependency）的建模能力。对于更复杂的金融数据，循环神经网络（RNN/LSTM）或 Transformer 架构可能会有更好的表现。此外，本实验仅使用了价格数据，未考虑成交量、新闻情绪等外部因素，这也是未来改进的方向。未来的工作可以尝试引入：

- **多模态数据：**结合财经新闻、社交媒体情绪等文本数据。
- **更复杂的架构：**使用 Transformer 或 Temporal Fusion Transformer (TFT) 捕捉长程依赖和变量间的交互。
- **强化学习：**直接优化交易策略的收益，而不是仅仅预测价格。

6 结论

本报告通过对比实验，深入分析了 MLP 在混沌系统和金融时间序列预测中的应用。实验结果表明：

1. MLP 能够有效学习洛伦兹系统的非线性动力学特征，实现高精度预测。

2. 在股票预测中，基于自身历史的单变量预测模型表现最优，优于引入关联股票的多变量模型。
3. 历史窗口大小对模型性能有显著影响。实验发现较短的历史窗口（如 5 天）能够取得更低的预测误差（ $MSE \approx 205.36$ ），而随着窗口增大，噪声干扰加剧，导致性能显著下降。
4. 与简单的 Persistence Model 相比，MLP 并没有表现出显著的优势，这揭示了金融时间序列预测的极高难度和随机游走特性。
5. Deep Ensembles 是一种简单有效的不确定性量化方法，能够直观地反映模型在不同任务下的置信度，为模型评估提供了多维度的视角。