



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

综合设计说明书

作者：张茂源 学号：921107810238

学院：经济管理学院

专业(方向)：计算机科学与技术（辅修）

题目：基于 Streamlit 的时序预测系统

指导者：贾修一

2025 年 5 月 29 日

综合设计说明书摘要

摘要：随着金融市场复杂性增加，传统时间序列预测方法在处理金融数据的非线性特征时存在局限性，现有预测工具缺乏友好交互界面且难以实现多方法比较。本研究设计并实现了基于 Streamlit 的多模型时间序列预测系统，集成 ARIMA 统计方法与 LSTM 深度学习技术，提供完整的数据预处理、特征工程、模型训练和性能评估 workflow。系统采用模块化设计，ARIMA 模块提供完整统计分析工具，LSTM 模块采用多阶段特征筛选策略解决高维特征问题。系统核心创新在于建立标准化多模型比较框架，通过数据对齐算法和多维度评估指标实现公平比较。基于 Apache ECharts 的交互式可视化提供丰富图表展示，显著提升用户体验。测试结果表明系统能有效处理中等规模时序数据，各环节表现稳定准确。本研究通过集成多种建模方法、降低技术门槛、提供标准化评估，为时序预测领域提供了有价值的工具平台，具有重要教育和实践价值。

关键词：Streamlit、时序预测、LSTM、ARIMA、模型评估、Web 应用、数据可视化

目 录

1 引言	1
1.1 研究背景与意义	1
1.2 研究目标与主要内容	2
1.2.1 研究目标	2
1.2.2 主要研究内容	2
1.3 技术路线与可行性分析	2
1.3.1 技术架构选择	2
1.3.2 模型选择的理论依据	3
1.3.3 技术可行性评估	3
1.4 论文结构安排	3
2 相关技术与理论基础	5
2.1 Streamlit 框架介绍	5
2.1.1 核心技术特性	5
2.1.2 多页面应用架构	5
2.1.3 交互机制与状态管理	5
2.2 时序预测模型	5
2.2.1 ARIMA 模型理论基础	6
2.2.2 LSTM 网络架构	7
2.2.3 模型比较与选择	8
2.2.4 其他时序预测模型	8
2.3 模型评估指标	9
2.3.1 基于误差的精度指标	9
2.3.2 金融时序特有评估指标	9
2.4 收益率预测方法	10
2.4.1 收益率预测的传统方法	10
2.4.2 基于深度学习的股价预测方法	10
2.5 数据可视化技术	11
2.6 技术栈核心	11
3 系统需求分析	13
3.1 功能性需求	13
3.2 非功能性需求	14
3.3 用户场景分析	14
4 系统设计	16
4.1 系统架构设计	16
4.1.1 整体架构模式	16
4.1.2 模块化组织结构	17

- 4.1.3 状态管理机制..... 17
- 4.2 数据处理模块设计..... 17
 - 4.2.1 数据导入与标准化..... 17
 - 4.2.2 技术指标计算引擎..... 17
 - 4.2.3 数据可视化框架..... 18
- 4.3 模型训练模块设计..... 19
 - 4.3.1 LSTM 深度学习框架..... 19
 - 4.3.2 ARIMA 统计建模框架..... 19
- 4.4 模型评估模块设计..... 22
 - 4.4.1 多模型比较框架..... 22
 - 4.4.2 评估报告生成系统..... 22
- 4.5 核心工具函数设计..... 22
 - 4.5.1 数据处理工具集..... 22
 - 4.5.2 图表生成工具集..... 22
 - 4.5.3 会话管理工具集..... 22
- 5 系统实现 23**
 - 5.1 技术栈与开发环境..... 23
 - 5.2 数据处理与预处理实现..... 23
 - 5.2.1 数据标准化机制..... 23
 - 5.2.2 技术指标计算引擎..... 23
 - 5.3 LSTM 深度学习模型实现..... 24
 - 5.3.1 特征工程与选择..... 24
 - 5.3.2 序列数据构造与归一化..... 24
 - 5.3.3 神经网络架构与训练..... 24
 - 5.4 ARIMA 统计建模实现..... 24
 - 5.4.1 时间序列预分析..... 24
 - 5.4.2 参数估计与模型拟合..... 25
 - 5.4.3 模型诊断与残差分析..... 25
 - 5.5 模型评估与比较实现..... 25
 - 5.5.1 数据对齐与指标计算..... 25
 - 5.5.2 可视化与报告生成..... 25
 - 5.6 会话状态管理实现..... 27
- 6 系统测试与分析 28**
 - 6.1 测试方法论与环境配置..... 28
 - 6.1.1 测试策略设计..... 28
 - 6.1.2 测试环境构建..... 28
 - 6.2 功能验证与性能评估..... 28
 - 6.2.1 数据处理模块验证..... 28

6.2.2 模型训练与评估验证.....	28
6.3 系统性能分析.....	29
6.3.1 计算性能评估.....	29
6.3.2 用户体验评估.....	29
6.4 缺陷识别与解决方案.....	29
7 总结与展望	30
7.1 研究成果总结.....	30
7.2 系统局限性分析.....	30
7.3 发展方向与改进策略.....	31
7.4 学术贡献与实践意义.....	31
7.5 结语.....	32
参考文献	33

1 引言

1.1 研究背景与意义

金融市场作为现代经济体系的核心组成部分，其价格发现机制和风险配置功能对宏观经济稳定具有重要意义。随着金融市场的日益复杂化和数据化，准确预测金融资产价格变动已成为学术界和实务界共同关注的重要课题^[1]。金融时间序列数据具有高频性、高维性和高噪声等特征，同时表现出显著的非线性、异方差性和时变性，这些特性使得传统的线性预测方法面临严峻挑战^[2]。金融资产价格的波动受到宏观经济基本面、市场微观结构、投资者行为偏差以及外部冲击等多重因素的综合影响，呈现出复杂的动态特征^[3]。有效的价格预测不仅对投资组合管理、风险控制和资产定价具有直接价值，更是金融市场有效性检验和货币政策传导机制研究的重要基础。传统的时间序列计量方法，如自回归积分滑动平均模型(ARIMA)、广义自回归条件异方差模型(GARCH)等，在金融时间序列建模中发挥了重要作用^[4]。这些方法基于严格的统计理论基础，具有良好的可解释性和理论支撑。然而，传统方法通常基于线性假设和特定的分布假定，在处理金融市场中普遍存在的非线性关系、结构性突变和复杂依赖结构时存在固有局限性^[5]。

近年来，深度学习技术在计算机视觉、自然语言处理等领域取得的突破性进展，为金融时间序列预测提供了新的研究范式^[2,6,7]。深度神经网络通过多层非线性变换，能够自动学习数据中的复杂模式和高阶特征交互，在处理高维、非线性和非平稳时间序列方面展现出显著优势^[8-10]。其中，长短期记忆网络(LSTM)等循环神经网络架构，通过门控机制有效解决了传统循环网络的梯度消失问题，在捕捉时间序列长期依赖关系方面表现突出^[7,11]。然而，深度学习方法也面临模型复杂度高、可解释性不足、对数据质量要求严格等挑战。因此，在实际应用中，传统计量方法与深度学习技术的有机结合，能够充分发挥各自优势，为金融时间序列预测提供更加稳健和全面的解决方案^[12]。

随着数据科学和机器学习技术的快速发展，构建用户友好的交互式分析平台已成为促进理论研究向实际应用转化的重要途径。传统的命令行工具和静态分析报告难以满足现代数据分析的交互性和实时性需求。基于 Web 的交互式平台能够为研究人员和实务工作者提供直观的数据探索、模型配置、训练监控和结果评估环境，显著降低了高级分析技术的使用门槛。此外，交互式平台还具有重要的教学和研究价值。通过可视化的界面和实时的反馈机制，用户能够更好地理解模型的工作原理、参数对结果的影响以及不同方法的适用场景，从而促进理论知识与实践应用的深度融合。

1.2 研究目标与主要内容

1.2.1 研究目标

本研究旨在设计并实现一个基于 Streamlit 框架的多模型时间序列预测系统，该系统集成传统计量方法与深度学习技术，为金融时间序列分析提供完整的建模 workflow。具体目标包括：

- (1) 构建涵盖数据预处理、特征工程、模型训练和性能评估的完整分析流程；
- (2) 实现 ARIMA 和 LSTM 等代表性预测模型的集成与比较；
- (3) 开发用户友好的交互式界面，支持参数配置、训练监控和结果可视化；
- (4) 建立标准化的模型评估体系，支持多维度性能比较和分析。

1.2.2 主要研究内容

本研究的核心内容包括以下几个方面：

1. 数据处理与特征工程模块：实现金融时间序列数据的标准化预处理流程，包括数据清洗、缺失值处理、异常值检测等基础功能。集成常用技术指标计算方法，如移动平均线、相对强弱指数(RSI)、移动平均收敛发散指标(MACD)、布林带等，为后续建模提供丰富的特征变量。设计交互式数据可视化组件，支持 K 线图、趋势图、相关性分析等多种展示方式。

2. 多模型预测框架：

(1) ARIMA 模型实现：基于经典时间序列理论，实现完整的 ARIMA 建模流程，包括平稳性检验、模型识别、参数估计和诊断检验。支持自动参数选择和手动参数配置两种模式，提供残差分析和预测区间估计功能。

(2) LSTM 模型实现：基于 PyTorch 框架构建深度学习预测模型，支持灵活的网络结构配置，包括隐藏层维度、网络深度、正则化参数等。实现特征选择机制，包括相关性分析、方差膨胀因子(VIF)检验和统计显著性检验。

3. 模型评估与比较系统：建立多维度的模型性能评估体系，包括均方误差(MSE)、均方根误差(RMSE)、平均绝对误差(MAE)、决定系数(R^2)和方向准确率等指标。设计可视化比较工具，支持预测结果对比、误差分析和模型诊断。实现评估报告的自动生成和导出功能。

4. 交互式用户界面：基于 Streamlit 框架设计多页面 Web 应用，提供直观的操作界面和实时反馈机制。实现会话状态管理，确保用户操作的连续性和数据的一致性。

1.3 技术路线与可行性分析

1.3.1 技术架构选择

本研究采用基于 Python 生态系统的技术架构，主要考虑因素如下：

第一，Streamlit 框架的优势。Streamlit 作为新兴的数据应用开发框架，具有开发效率高、学习成本低、部署便捷等特点。其基于 Python 的纯代码开发模式与数据科学工作流高度契合，

能够快速将研究原型转化为可交互的应用系统。框架提供的丰富组件库和自动化状态管理机制，显著简化了复杂交互逻辑的实现。

第二，Python 生态系统的完备性。Python 在数据科学和机器学习领域拥有成熟完善的生态系统。NumPy 和 Pandas 提供了高效的数值计算和数据处理能力；Statsmodels 实现了完整的统计建模功能；PyTorch 提供了灵活的深度学习框架；Matplotlib、Plotly 和 ECharts 等可视化库支持丰富的图表展示需求。这些库的有机结合为本研究提供了坚实的技术基础。

1.3.2 模型选择的理论依据

ARIMA 模型作为时间序列分析的经典方法，具有坚实的理论基础和广泛的应用验证。其线性建模框架虽然相对简单，但在处理具有明确趋势和季节性特征的时间序列时仍具有良好的预测性能。更重要的是，ARIMA 模型的可解释性强，能够为复杂的深度学习模型提供基准比较。LSTM 网络通过门控机制有效解决了传统循环神经网络的梯度消失问题，在处理长序列依赖关系方面具有显著优势。其非线性建模能力使其能够捕捉金融时间序列中的复杂模式，特别适合处理多变量、高频和非平稳的金融数据^[7]。ARIMA 和 LSTM 代表了时间序列预测的两种不同范式——统计建模和机器学习。两者的结合能够充分发挥各自优势，为不同特征的数据和应用场景提供最适合的建模方案。

1.3.3 技术可行性评估

本研究所选择的技术栈均为开源项目，拥有活跃的社区支持和完善的文档体系。相关库的 API 稳定，版本兼容性良好，为项目的顺利实施提供了保障。与此同时，ARIMA 模型的计算复杂度相对较低，普通计算环境即可满足需求；LSTM 模型虽然计算量较大，但通过合理的模型设计和参数配置，在 CPU 环境下也能实现有效训练。如有 GPU 资源，可进一步提升训练效率。进一步地，本研究采用模块化的系统设计和标准化的代码规范，确保系统具有良好的可扩展性。此外，多页面应用架构为后续功能扩展提供了灵活的框架支持。最后，Web 界面的直观性和交互性能够有效降低用户的学习成本，提高系统的实用价值和推广潜力。综上所述，本研究在技术上具有较强可行性。

1.4 论文结构安排

本文的组织结构如下：第一章：引言。主要介绍项目的研究背景、意义、目标、主要研究内容以及技术选型和可行性。第二章：相关技术与理论基础。详细阐述 Streamlit 框架、时间序列预测模型以及相关的评估方法等。第三章：系统需求分析。从功能需求、非功能需求和用户场景等方面对系统进行全面分析。第四章：系统设计。阐述系统的总体架构、各主要功能模块的设计思路 and 核心工具函数设计。第五章：系统实现。具体描述开发环境、数据处理流程、LSTM 和 ARIMA 模型的具体实现细节、模型评估与报告生成的实现方式。第六章：系统测试与分析。介绍测试策略、功能测试和非功能测试的执行情况，分析测试结果和遇到的

问题，并对未来测试工作进行展望。第七章：总结与展望。对整个项目进行总结，归纳主要成果和创新点，分析存在的不足，并对未来的改进方向和扩展功能进行展望。

2 相关技术与理论基础

2.1 Streamlit 框架介绍

Streamlit 是一个基于 Python 的开源 Web 应用框架，专门为数据科学和机器学习应用的快速开发而设计。该框架采用声明式编程范式，允许开发者使用纯 Python 代码构建交互式 Web 应用，使得前端开发变得高效。

2.1.1 核心技术特性

Streamlit 的技术架构基于脚本重新运行机制，即当用户与界面组件交互时，整个 Python 脚本会重新执行，并智能更新发生变化的界面元素。这种设计简化了状态管理的复杂性，使开发者能够专注于业务逻辑的实现。框架提供了内置的缓存机制，通过装饰器形式的缓存策略有效避免重复计算，提升应用性能。

在组件体系方面，Streamlit 提供了完整的用户界面组件库，涵盖数据展示、用户输入和页面布局三个层面。数据展示组件支持多种数据类型的智能渲染，包括文本、表格、图表和媒体文件。用户输入组件提供了丰富的交互控件，如滑块、选择框、按钮等，支持实时数据绑定。页面布局组件通过列布局、侧边栏、标签页等方式实现复杂界面的组织。

2.1.2 多页面应用架构

Streamlit 的多页面应用架构采用约定优于配置的设计理念。通过在项目根目录下创建 pages 子目录，并将各功能模块的 Python 脚本放置其中，框架会自动识别并生成导航结构。这种架构模式支持功能模块的独立开发和维护，同时通过 session_state 机制实现跨页面的数据共享和状态管理。本研究采用该架构模式，将数据处理、模型训练和模型评估三个核心功能分别实现为独立页面，既保证了功能的模块化，又确保了用户操作的连贯性。

2.1.3 交互机制与状态管理

Streamlit 的交互机制基于响应式编程模型，用户的每次交互都会触发脚本的重新执行。为了在多次执行间保持数据状态，框架提供了 session_state 对象，支持复杂数据结构的持久化存储。这种机制特别适合数据分析应用中的中间结果保存和跨步骤的数据传递。

2.2 时序预测模型

时间序列预测是计量经济学和机器学习的重要研究领域，涉及从历史观测数据中学习时间依赖模式，并据此预测未来值。本研究选择 ARIMA 和 LSTM 作为代表性方法，分别体现了统计建模和深度学习两种不同的建模范式。

2.2.1 ARIMA 模型理论基础

ARIMA(p,d,q)模型是由 Box 和 Jenkins 于 20 世纪 70 年代提出的差分自回归移动平均模型(Autoregressive Integrated Moving Average Model), 是时间序列分析的经典方法。该模型通过自回归(AR)、差分(I)和移动平均(MA)三个组成部分的有机结合, 能够有效处理具有趋势性、季节性和自相关性的非平稳时间序列数据。

自回归模型 AR(p)利用时间序列自身的历史值与当前值之间的线性关系建立预测模型。若时间序列 $\{Y_t\}$ 的当前值可表示为其滞后项的线性组合, 则有:

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{t-p} + \varepsilon_t$$

其中, φ_i 为自回归系数, p 为自回归阶数, $\{\varepsilon_t\}$ 为满足零均值、同方差且无自相关的白噪声序列。

AR 模型的平稳性条件通过特征方程确定。引入滞后算子 L , 定义 $LY_t = Y_{t-1}$, 则 AR(p)模型的特征方程为:

$$1 - \varphi_1 L - \varphi_2 L^2 - \cdots - \varphi_p L^p = 0$$

当特征方程所有根的模长均大于 1 时, AR(p)过程为平稳过程。

移动平均模型 MA(q)模型通过当前及历史扰动项的线性组合构建预测模型:

$$Y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

其中, θ_i 为移动平均系数, q 为移动平均阶数。MA 模型天然具有平稳性, 其可逆性条件要求特征方程 $1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q = 0$ 的所有根模长大于 1。

ARIMA(p,d,q)模型将 AR 和 MA 模型相结合, 并通过 d 阶差分处理非平稳序列:

$$\varphi(L)(1-L)^d Y_t = c + \theta(L)\varepsilon_t$$

其中,

$$\varphi(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p$$

为自回归多项式,

$$\theta(L) = 1 + \theta_1 L + \cdots + \theta_q L^q$$

为移动平均多项式, $(1-L)^d$ 为 d 阶差分算子。

ARIMA 的建模包括模型识别、参数估计和模型诊断三个步骤。模型识别主要依赖自相关函数(ACF)和偏自相关函数(PACF)的分析模式: AR(p)模型的 PACF 在 p 阶后截尾, ACF 呈指数衰减; MA(q)模型的 ACF 在 q 阶后截尾, PACF 呈指数衰减。参数估计通常采用极大似然估计或最小二乘估计方法。模型诊断通过残差分析验证模型充分性, 要求残差序列为白噪声过程。

平稳性检验是 ARIMA 建模的前提条件。常用的单位根检验方法包括增广迪基-富勒检验(ADF)和菲利普斯-佩伦检验(PP)。对于非平稳序列, 通过适当阶数的差分操作实现平稳化, 差分阶数 d 的确定需要平衡平稳性要求与信息损失的权衡。

2.2.2 LSTM 网络架构

LSTM 网络是循环神经网络的重要变体，专门设计用于解决传统 RNN 在处理长序列时面临的梯度消失问题。其核心创新在于引入了门控机制，通过遗忘门、输入门和输出门的协调控制，实现对信息的选择性记忆和遗忘。

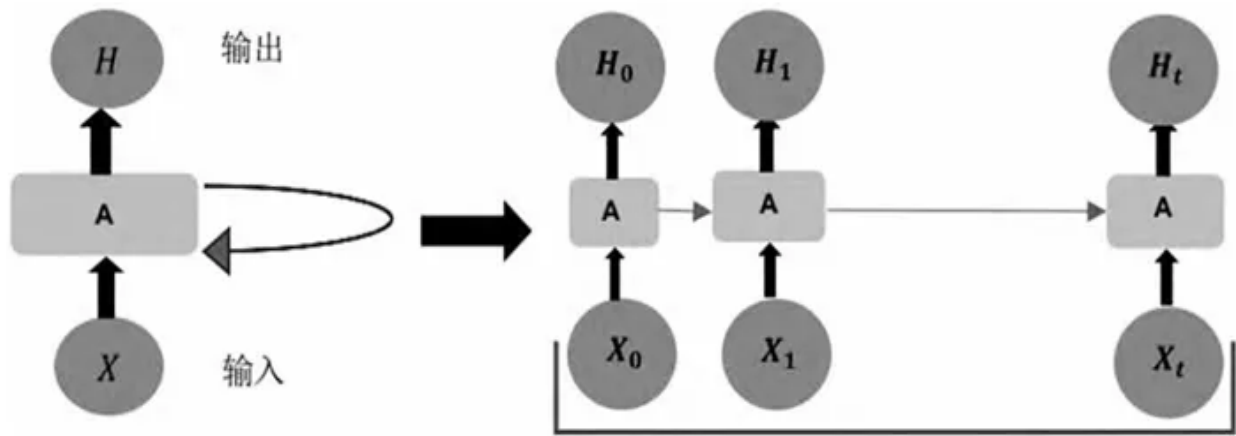


图 2.1 结合之前状态的 RNN 结构图

LSTM 的基本单元结构如图所示，包含三个关键的门控机制：遗忘门决定从细胞状态中丢弃哪些信息，输入门确定哪些新信息将被存储在细胞状态中，输出门控制基于细胞状态输出哪些部分。这种精巧的门控设计使得 LSTM 能够有效地学习和记忆长期依赖关系。

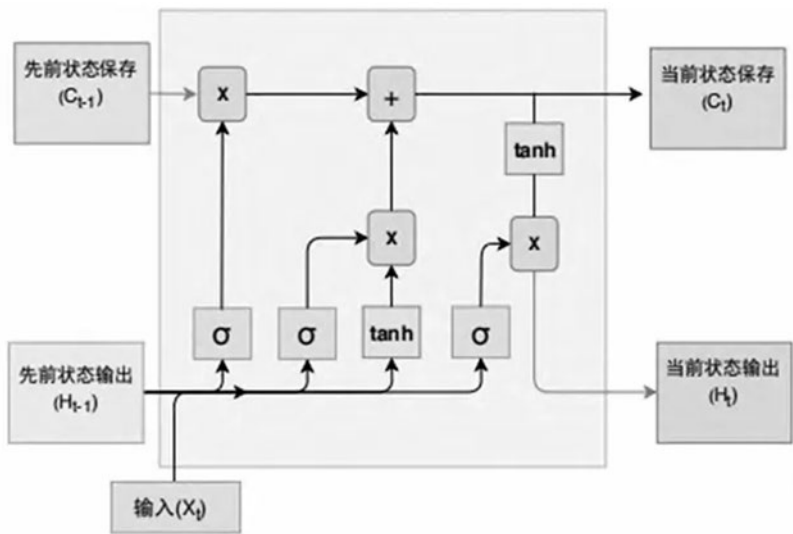


图 2.2 基本的 LSTM 单元结构

LSTM 的细胞状态机制是其处理长期依赖关系的关键。细胞状态作为信息的载体，在时间步之间传递，而门控结构则决定哪些信息应该被保留、更新或输出。这种设计使得 LSTM 能够在长序列中保持重要信息，同时过滤掉无关的噪声。

在时间序列预测任务中，LSTM 通过滑动窗口方法将时间序列转换为监督学习问题。输入序列的长度和特征维度是影响模型性能的重要超参数。数据预处理通常包括归一化操作，以确保不同尺度的特征能够得到平衡的学习。

2.2.3 模型比较与选择

ARIMA 和 LSTM 代表了时间序列建模的两种不同范式。ARIMA 基于严格的统计理论，具有良好的可解释性和理论基础，适合处理具有明确统计特征的时间序列。LSTM 则依赖数据驱动的学习机制，能够自动发现复杂的非线性模式，特别适合处理高维、多变量的时间序列数据。在模型选择方面，需要综合考虑数据特征、计算资源和预测目标等因素。对于具有明确趋势和季节性的序列，ARIMA 通常能够提供稳定的预测性能。而对于包含复杂非线性关系的序列，LSTM 的表现往往更为优越。在实际应用中，两种方法的结合使用能够充分发挥各自优势，为不同场景提供最适合的建模方案。

2.2.4 其他时序预测模型

除 ARIMA 和 LSTM 外，时间序列预测领域还涉及多种其他各具特色和适用场景的建模方法。传统的循环神经网络(RNN)是序列建模的基础架构，通过隐藏状态的循环连接实现时间信息的传递。然而，RNN 在处理长序列时面临梯度消失问题，限制了其在复杂时序任务中的应用。反向传播神经网络(BP)作为前馈网络的代表，虽具备非线性建模能力，但缺乏处理序列依赖关系的内在机制，在时序预测中的直接应用受到限制^[13]。卷积神经网络(CNN)通过局部感受野和权值共享机制，能够有效提取时间序列的局部模式特征。在金融时序分析中，CNN 常被用于处理高频数据或将时序数据转换为图像形式进行模式识别^[14,15]。Transformer 模型基于自注意力机制，能够并行处理序列信息并捕捉长距离依赖关系，在处理多变量、长序列的时序预测任务中展现出良好的潜力^[16-18]。这些模型在特定应用场景下各有优势，但考虑到本研究的数据特征和计算资源约束，选择 ARIMA 和 LSTM 作为核心建模方法能够在模型复杂度和预测性能之间取得良好平衡。

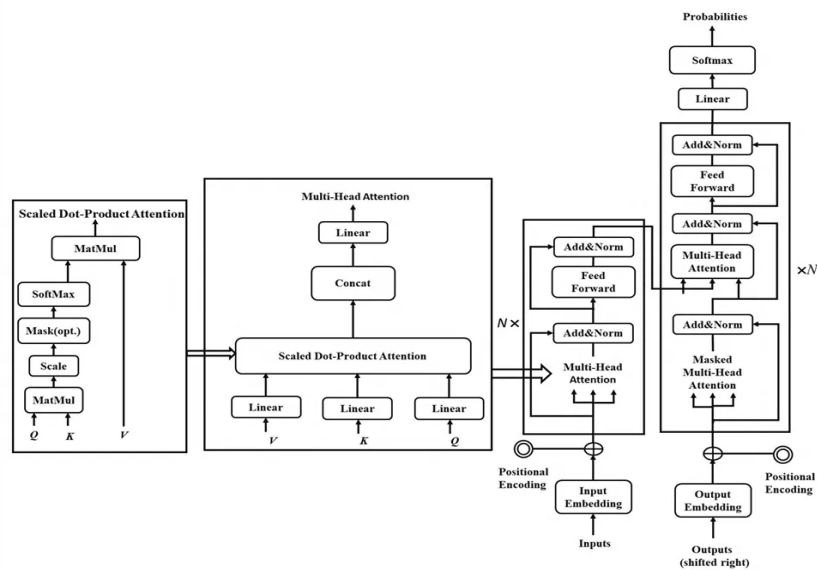


图 2.3 基于 Transformer 的时间序列预测原理示意图

2.3 模型评估指标

时间序列预测模型的性能评估是确保模型有效性和实用性的关键环节。合理的评估指标体系不仅能够量化模型的预测精度，更能从多维度反映模型的泛化能力、稳健性和实际应用价值。

2.3.1 基于误差的精度指标

时间序列预测本质上属于回归问题范畴，因此经典的回归评估指标在此领域得到广泛应用。设真实值序列为 $\{y_t\}_{t=1}^n$ ，预测值序列为 $\{\hat{y}_t\}_{t=1}^n$ ，则主要精度指标定义如下：

均方误差 (MSE) 通过平方损失函数衡量预测偏差，对大误差具有惩罚放大效应：

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

均方根误差 (RMSE) 作为 MSE 的平方根，具有与原始数据相同的量纲，便于直观解释：

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

平均绝对误差 (MAE) 采用绝对值损失，对异常值的敏感性低于 MSE，能更好地反映预测误差的典型水平：

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

拟合优度 (R^2) 衡量模型对数据变异的解释能力，取值范围为 $(-\infty, 1]$ ，越接近 1 表示拟合效果越佳：

$$R^2 = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y})^2}$$

其中 $\bar{y} = \frac{1}{n} \sum_{t=1}^n y_t$ 为样本均值。

2.3.2 金融时序特有评估指标

金融时间序列预测中，价格变动方向的准确性往往比绝对数值精度更具实际意义，因此需要引入专门的方向性指标和相对误差度量：

方向准确率 (Directional Accuracy) 评估模型对价格变动趋势的预测能力：

$$DA = \frac{1}{n-1} \sum_{t=2}^n \mathbb{I}[(\hat{y}_t - y_{t-1})(y_t - y_{t-1}) > 0]$$

其中 $\mathbb{I}[\cdot]$ 为示性函数。该指标反映模型捕捉市场趋势转换的能力，在金融投资决策中具有重要的实用价值。

平均绝对百分比误差 (MAPE) 提供相对误差度量，便于跨不同价格水平的序列比较：

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

需注意当 y_t 接近零时该指标可能失效。

异质调整误差指标为了更好地处理金融时序数据的异方差特性，引入异质调整评估指标，这些异质调整指标能够更准确地反映模型在不同市场环境下的预测稳定性，特别适用于具有时变波动率特征的金融时间序列。其中，异质调整平均绝对误差(HMAE)通过标准化处理消除不同时期波动率差异的影响：

$$HMAE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

异质调整均方误差(HMSE)在平方损失基础上进行相对化调整：

$$HMSE = \frac{1}{n} \sum_{t=1}^n \left(\frac{y_t - \hat{y}_t}{y_t} \right)^2$$

参考现有研究，本文采用 MSE、RMSE、MAE、 R^2 、方向准确率和 MAPE 构成多维度评估体系，以全面衡量模型在精度、方向性预测和相对误差控制方面的综合表现^[19]。该评估框架既考虑了统计意义上的预测精度，又兼顾了金融应用中的实际需求。

2.4 收益率预测方法

2.4.1 收益率预测的传统方法

如何对金融市场价格进行准确预测，一直是一个经典难题。由于金融市场的复杂性、高噪声和高波动性，及时预测金融市场走势具有很大的难度。金融市场价格预测问题的根源来自于对于有效市场假说(EMH)的争论。Fama 提出：市场是有效的，任何关于资产价格的技术分析和基本面分析都是无效的^[20]。许多学者对此提出质疑，他们通过传统计量模型如自回归模型(AR)、自回归条件异方差模型(ARCH)^[21]、随机游走模型(Random Walk Theory)等以及机器学习的方法，建立线性模型通过回归得出了关于反对市场有效性的结论。

然而，伴随着数据的激增，金融市场变得更加复杂，金融数据中包含更多噪声以及更强不确定性，传统方法似乎不再奏效：依赖于严格假定的传统的参数模型，往往难以捕捉金融市场日益复杂的非线性因素；而传统的非参数估计虽然不需要对未知函数进行一致性估计，但在处理高维数据时却存在“维数灾难”问题；传统机器学习虽然可以灵活地处理高维数据，但仍存在过度依赖人工设计、过拟合、收敛速度慢等缺陷^[22,23]。

2.4.2 基于深度学习的股价预测方法

随着机器学习和人工智能算法的快速发展，与传统方法相比，基于深度学习的预测方法更有利于分析和预测具有非线性和波动性的股票变化趋势。近年来的相关实证研究逐渐展现了深度学习在解决金融市场价格预测问题上的优势。

针对股票预测，RNN 及其变种成为最优先选择的深度学习模型。2018 年，Hajiabotorabi 等^[24]采用优化的 RNN 预测了 4 种不同的股票价格。为了解决高噪声股票市场数据没有可学习的监督信息的问题，并保证交易决策的鲁棒性，Si 等采用四层全连接网络和 LSTM 对中国

股市进行预测，通过不同权重的目标对模型进行训练，获得了更好的预测效果^[10]。2022 年章宁^[19]等人使用 Transformer 模型进行预测，并将预测结果与 LSTM 及 SVR 模型进行对比，取得了预期超额收益率预测的良好效果。

单一深度学习方法有其自身的优缺点。如 DMLP 具有解决复杂非线性模式的能力，但局部最小化会影响其训练过程；LSTM 虽然能够长时间记住信息并分析数据中的交互作用和隐藏模式，但在写入和读取数据时缺乏索引，计算费时。因此，部分研究尝试将两个或多个深度学习模型融合，以结合单一模型的优点构建新模型来改善性能。现有的许多研究已经表明集成学习方法比单一深度学习模型具有更好的性能。2018 年 Zhou 等人提出了一个利用 LSTM 和卷积神经网络(CNN)进行对抗性训练来预测高频股市的通用框架^[25]。2018 年谷歌(Google)开发了一种双向的 Transformer 框架 BERT (Bidirectional Encoder Representations from Transformers)^[8]。2021 年 Lu 等人也是将 CNN、双向长短期记忆神经网络 (BiLSTM) 与注意机制 (AM) 进行了结合，应用到上证综合指数上^[14]。2022 年 Darapaneni 等人使用 LSTM 与随机森林进行混合搭建，利用历史价格和可用的情绪数据来预测股票的未来走势^[26]。2022 年 Shi 等人提出一种基于注意力机制的 CNN-LSTM 和极限梯度提升算法 (XGBoost) 混合模型来预测股价^[27]。2024 年危冰淋等人提出一种 Transformer-LSTM 多因素碳排放权交易价格预测的深度学习模型，同时运用 SVR、MLP、LSTM、Transformer 单一模型进行预测，结果表明 Transformer-LSTM 模型得到的预测价格与湖北省碳排放权交易价格 (HBEA) 的实际价格更为吻合^[16]。

2.5 数据可视化技术

数据可视化在时间序列分析与模型评估中占据核心地位，其不仅辅助研究者洞察数据内在模式与模型行为，亦是评估与传达模型性能的关键手段。本研究选用 Apache ECharts 作为主要的图表渲染引擎，使用 streamlit-echarts 库将其集成于 Streamlit 应用中。ECharts 具备丰富的图表类型（如 K 线图、热力图、雷达图等）、高度的交互性（如数据缩放、提示框、图例切换）以及灵活的配置机制，能够满足本研究多样化和精细化的可视化需求。例如，在数据探索阶段，利用 ECharts 生成交互式 K 线图、成交量图及技术指标相关性热力图；在模型训练与评估阶段，则用于损失函数曲线、自相关与偏自相关图谱、预测序列与实际序列的对比图、以及模型性能的多维度对比雷达图与柱状图的动态展示。

2.6 技术栈核心

本研究的技术实现依托于 Python 编程语言及其成熟的数据科学与机器学习生态系统。核心依赖库及其在项目中的作用如下：Pandas 与 NumPy 构成了数据处理与数值计算的基石。Pandas 提供的高性能 DataFrame 结构为时间序列数据的加载、清洗、转换、索引及技术指标计算提供了便利；NumPy 则为底层的多维数组运算提供了高效支持，并广泛应用于 Pandas 及深度学习框架的数据操作中。Scikit-learn (Sklearn) 作为综合性的机器学习库，在本研究中主

要承担数据预处理（如特征归一化）、模型评估指标计算（如 MSE, MAE, R^2 ）以及辅助特征选择（如统计显著性检验）等任务。Statsmodels 专注于统计建模与推断，是实现 ARIMA 模型的关键。其不仅提供了 ARIMA 模型的拟合、预测与诊断功能，还包含了时间序列分析所需的多种统计检验工具（如 ADF 检验、Ljung-Box 检验）和 ACF/PACF 图的计算基础。PyTorch 作为主流的深度学习框架，以其灵活性与动态计算图特性，被用于构建、训练和评估 LSTM 神经网络模型。这包括定义网络结构（LSTM 层、全连接层）、选择损失函数与优化器，以及利用张量运算实现高效的 GPU 加速训练。

上述库的协同工作，为本研究从数据准备到复杂模型构建、训练及精细化评估提供了全面的技术支撑。

```
1  # Web应用框架
2  streamlit>=1.24.0
3  streamlit-echarts>=0.4.
4  0
5  # 数据处理和分析
6  numpy==1.26.4
7  pandas>=1.5.3
8  scipy>=1.10.0
9
10 # 深度学习框架
11 torch>=2.0.1
12 torchvision>=0.10.0
13 torchaudio>=0.9.0
14
15 # 数据可视化
16 plotly>=5.18.0
17 matplotlib>=3.7.1
18 seaborn>=0.12.2
19
20 # 机器学习和评估
21 scikit-learn>=1.2.2
22
23 # 技术分析指标
24 ta>=0.10.0
```

图 2.4 部分依赖库目录

3 系统需求分析

本章旨在明确界定基于 Streamlit 的多模型时间序列预测系统的核心需求。系统构建围绕三大核心功能模块展开：数据探索与预处理、模型构建与训练、以及模型效能评估与比较。这些模块通过主界面实现导航与状态信息共享。

3.1 功能性需求

系统的功能性需求聚焦于为用户提供一个完整、高效的时间序列分析与预测 workflow。

数据探索与预处理模块 (1_DataView.py) 需支持 CSV 格式时序数据的灵活导入，包括用户上传与内置示例数据的加载。关键功能点在于对数据列（如日期、价格、成交量）的智能识别与标准化映射，确保数据按时间正确排序，并提供基础统计描述。该模块的核心在于技术指标的计算与可视化，用户可选择计算多种常用指标（如移动平均线、相对强弱指数、MACD、布林带等），并通过交互式图表（如 K 线图、成交量图、相关性热力图）进行深入探索。处理后的原始数据及技术指标数据均支持导出。

模型构建与训练模块 (2_ModelTraining.py) 以支持 LSTM 与 ARIMA 两种主流预测模型为核心。页面左侧的通用配置区域涵盖训练集与测试集的划分、已训练模型的持久化保存，以及关键评估指标的实时概览。针对 LSTM 模型，模块需提供灵活的特征选择机制，允许用户从原始数据或计算的技术指标中选取输入特征，并支持基于统计显著性（如相关性、VIF、P 值）的自动化特征筛选。超参数配置（如序列长度、隐藏层规模、层数、Dropout 率、学习率、批次大小及训练周期）需高度可定制化，并辅以早停机制。训练过程的可视化（如损失函数曲线）与训练完成后的多维度结果展示（包括主要评估指标、预测值与真实值对比图、预测散点图及误差分析图）是关键需求。针对 ARIMA 模型，模块需支持对单一目标序列进行必要的预处理（如对数变换、差分操作）与详尽的统计特性分析（包括平稳性检验、白噪声检验、ACF/PACF 图谱分析），以辅助用户进行模型定阶。参数配置支持手动设定 (p,d,q) 阶数与自动寻优（基于 AIC/BIC 等信息准则）两种模式，并允许选择动态或静态预测方式。为评估模型稳定性，系统应支持多次重复训练并依据特定指标选优。训练结束后的结果展示应包括核心评估指标、预测值与真实值对比图，以及详尽的残差分析（如残差序列图、分布图）。

模型效能评估与比较模块 (3_ModelEvaluation.py) 旨在提供对已训练模型的全面评估与横向比较。模块首先对会话中所有已成功训练的模型（LSTM、ARIMA）进行状态概览与初步性能排序。用户可通过侧边栏灵活选择参与比较的模型、关注的评估指标（MSE, RMSE, MAE, MAPE, 方向准确率, R^2 等）以及图表显示参数。核心功能包括：多模型在各项选定指标上的性能对比表格；基于多维度指标的雷达图，直观展现模型的综合性能；针对各单一指标的柱状图，高亮不同模型的优劣。预测结果的深入分析涵盖了在同一图表中对齐展示各模型预测序列与真实值，以及针对每个模型的“实际值 vs. 预测值”散点图。此外，系统需支持生成结构化的评估报告，用户可自定义报告包含的章节（如执行摘要、性能指标详情、预测

分析等）与输出格式（如 HTML 预览、Markdown 文本、JSON 数据），并提供报告文件及相关评估数据的下载功能。

3.2 非功能性需求

非功能性需求主要关注系统的性能、易用性、可扩展性及可靠性。

性能方面，要求中等规模数据集（数万行级别）的数据加载与技术指标计算能在数秒内完成。模型训练时长需在可接受范围内，LSTM 模型应提示并支持 GPU 加速，ARIMA 模型的自动参数寻优过程也应有合理的效率。界面交互需保持流畅，避免因计算或渲染导致的明显卡顿。

易用性是本系统的核心考量。界面设计需遵循直观、简洁的原则，确保用户通过少量操作即可完成主要分析流程。系统应提供清晰的操作指引与参数解释（如 tooltips），对潜在错误（如文件格式错误、数据缺失）进行友好提示与妥善处理。模型参数应提供合理的默认值，以降低初学者门槛。

可扩展性体现在模型库、功能模块及技术指标三个层面。系统架构设计应便于未来集成更多新兴的预测模型，方便在现有页面中增添新的分析工具或可视化维度，并易于扩展技术指标的计算方法。

可靠性与准确性要求数据处理（如指标计算、数据变换）的精确无误，LSTM、ARIMA 模型的实现严格遵循其标准算法，各项评估指标的计算准确。用户会话中的数据、参数配置及模型状态需通过 `st.session_state` 进行有效持久化，防止意外丢失。代码需具备良好的健壮性，能处理异常输入与边界条件。

3.3 用户场景分析

为具体阐释系统功能与价值，设定以下典型用户场景：

1. 数据探索与快速洞察：金融分析初学者或学生上传个人关注的股票数据，通过数据查看模块快速掌握数据概貌，审视 K 线图与成交量，计算并观察核心技术指标，完成初步的探索性数据分析。

2. ARIMA 模型应用实践：用户在掌握基础时序知识后，欲实践 ARIMA 模型。其在模型训练模块选择目标序列，执行必要的预处理与统计诊断（如平稳性检验、ACF/PACF 分析），随后可借助自动参数寻优或手动设置(p,d,q)参数完成模型训练，并细致审查预测结果与残差特性。

3. LSTM 模型参数调优：具备一定机器学习背景的用户倾向于使用 LSTM 进行预测。用户在模块内精心挑选输入特征，细致调整 LSTM 网络结构参数（如层数、隐藏单元数、学习率等），启动训练并实时监控损失曲线变化。训练完成后，重点关注各项评估指标、预测序列与真实序列的拟合程度。

4. 跨模型效能综合评估：用户在分别构建 LSTM 与 ARIMA 模型后，期望对其预测能力进行横向比较。此时，模型评估模块将提供全面的支持，用户可选定模型，在统一标准下（如相同测试集、相同评估指标）审视其性能数据（通过表格、雷达图、柱状图），并对比预测曲线的优劣，最终择优采纳，并生成详尽的评估报告。
5. 迭代优化与参数敏感性探索：当用户对初步模型结果不甚满意时，可返回模型训练模块，针对性调整关键参数（如 LSTM 的序列长度、ARIMA 的差分阶数），重新执行训练，再至评估模块观察调整后的性能变化，通过此类迭代优化过程，逐步逼近最优模型配置。

LSTMARIMAProphet

特征选择

筛选阈值设置

相关性阈值

0.50

1.00

1.00

VIF阈值

10.00

20.00

0.00

P值阈值

0.05

0.10

特征筛选

筛选特征

特征筛选完成！从 22 个特征中选出 5 个特征：['Lower_Band', 'MA10', 'MA5', 'MA20', 'Close']

选择训练特征

选择用于训练的特征

Lower_Band ×

MA10 ×

MA5 ×

MA20 ×

Close ×

×

▼

筛选详细结果

相关性筛选

VIF筛选

统计显著性筛选

模型参数

输入序列长度

20

预测序列长度

1

训练轮数

100

批次大小

32

隐藏层大小

64

LSTM层数

2

学习率

0.0010

Dropout比例

0.20

图 3.1 LSTM 模型参数设定界面

4 系统设计

本章从系统架构、功能模块和核心组件三个层面阐述基于 Streamlit 的多模型时序预测系统的设计理念与实现方案。系统采用模块化设计思想，通过清晰的职责分离和标准化的接口设计，确保各功能组件的高内聚、低耦合特性。

4.1 系统架构设计

4.1.1 整体架构模式

系统采用基于 Streamlit 框架的多页面 Web 应用架构，遵循 Model-View-Controller 设计模式的变体。应用入口通过 Home.py 提供系统概览和导航功能，核心业务逻辑分布在三个主要页面模块中：数据探索与预处理(1_DataView.py)、模型构建与训练(2_ModelTraining.py)以及模型评估与比较(3_ModelEvaluation.py)。



图 4.1 模型训练界面

4.1.2 模块化组织结构

系统采用分层架构设计，将业务逻辑、数据处理和用户界面有效分离。`src/models/`目录封装了 LSTM 和 ARIMA 模型的核心算法实现，包括特征工程、模型训练和评估逻辑。`src/utils/`目录提供通用工具函数，涵盖数据处理(`data_processing.py`)、图表生成(`chart_utils.py`)和会话状态管理(`session.py`)等功能。这种模块化设计不仅提高了代码的可维护性，也为后续功能扩展提供了良好的架构基础。

4.1.3 状态管理机制

系统通过 Streamlit 的 `session_state` 机制实现跨页面的数据持久化和状态同步。关键数据对象包括原始数据集、技术指标计算结果、模型训练参数、训练完成的模型实例以及评估指标等，均通过标准化的键值对形式存储在会话状态中。为确保状态管理的一致性和可靠性，系统在 `src/utils/session.py` 中封装了标准化的状态访问接口，包括安全的状态获取、设置和批量更新功能。

4.2 数据处理模块设计

4.2.1 数据导入与标准化

数据处理模块承担时序数据的接入、清洗和标准化任务。系统支持 CSV 格式的数据上传，并通过智能列名映射机制处理中英文混合或非标准命名的数据列。`normalize_column_names()`函数维护了完整的列名映射字典，确保数据能够正确映射到系统预期的标准格式(Date, Open, High, Low, Close, Volume)。

4.2.2 技术指标计算引擎

技术指标计算基于 TA-Lib 库的 Python 封装实现，支持包括趋势类(移动均线、MACD)、动量类(RSI、随机指标)、波动率类(布林带、ATR)和成交量类(OBV、成交量加权平均价)在内

选择训练特征

选择用于训练的特征

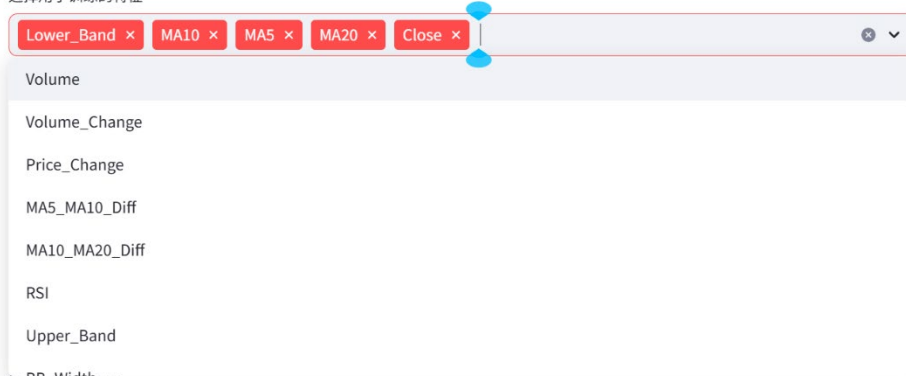


图 4.2 特征筛选

的多种指标计算。系统采用容错设计，当某些指标因数据不完整而无法计算时，能够降级并提供部分可用指标。计算结果以 DataFrame 形式组织，便于后续的特征选择和模型训练使用。

4.2.3 数据可视化框架

可视化模块基于 Apache ECharts 构建，通过 streamlit-echarts 库实现与 Streamlit 的无缝集成。系统设计了标准化的图表配置生成函数，包括 K 线图与成交量组合图(create_echarts_kline_volume)、相关性热力图(create_correlation_heatmap)等。所有图表配置均包含完整的交互功能(缩放、提示、图例控制)和导出功能，为用户提供专业级的数据分析体验。

特征相关性矩阵

选择要包含在相关性矩阵中的指标

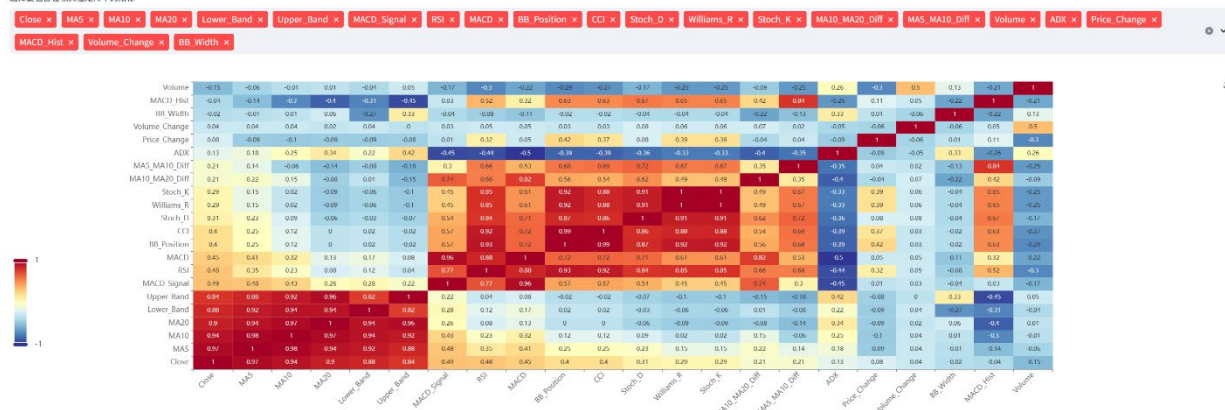


图 4.4 特征相关性热力矩阵

股票走势与成交量分析



图 4.3 股票日 K 线图

4.3 模型训练模块设计

4.3.1 LSTM 深度学习框架

LSTM 模型模块基于 PyTorch 框架构建，采用面向对象的设计模式。LSTMModel 类继承自 nn.Module，封装了多层 LSTM 网络结构、全连接输出层和前向传播逻辑。模型支持灵活的超参数配置，包括隐藏层维度、网络深度、Dropout 正则化率等关键参数。

特征选择机制是 LSTM 模块的核心创新点，系统实现了基于统计显著性的多阶段特征筛选流程。首先通过皮尔逊相关系数进行初步筛选，然后采用方差膨胀因子(VIF)检测并消除多重共线性，最后通过 F 检验的 P 值进行统计显著性验证。这种渐进式筛选策略既保证了特征的预测能力，又避免了维度灾难问题。

训练过程采用标准的监督学习范式，支持训练集/验证集划分、早停机制和学习率调度。系统通过滑动窗口技术将时间序列转换为监督学习问题，并采用 MinMaxScaler 进行数据归一化处理。训练过程的损失曲线和评估指标通过 Streamlit 组件实时展示，为用户提供直观的训练监控体验。

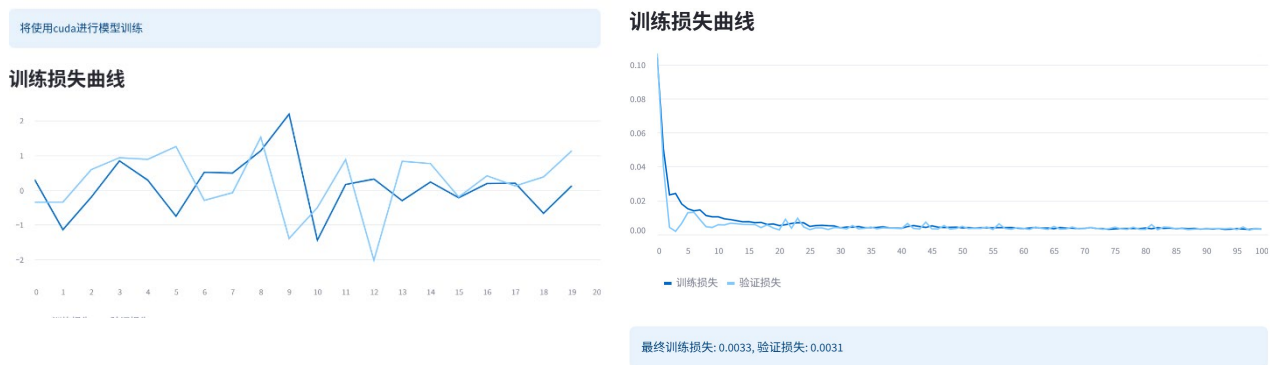


图 4.5 LSTM 模型训练损失曲线

4.3.2 ARIMA 统计建模框架

ARIMA 模型模块基于 Statsmodels 库实现，严格遵循 Box-Jenkins 建模方法论。系统提供了完整的时间序列预分析工具，包括平稳性检验(ADF 检验)、白噪声检验(Ljung-Box 检验)和自相关分析(ACF/PACF)。这些分析工具不仅为模型定阶提供统计依据，也帮助用户深入理解数据的时序特征。

参数确定支持自动寻优和手动配置两种模式。自动寻优通过网格搜索在用户指定的参数空间内寻找最优的(p,d,q)组合，优化目标基于 AIC 或 BIC 信息准则。手动配置模式则允许有经验的用户直接指定参数，适用于对数据特征有深入了解的场景。

模型诊断是 ARIMA 模块的重要组成部分，系统通过残差分析验证模型的充分性。残差序列的时序图、分布图和自相关图为用户提供了全面的模型诊断信息，确保模型满足 ARIMA 的基本假设条件。

数据预处理

选择需要分析的变量

Close

数据处理方法

- ☐ 原始数据
- ☒ 对数变换
- ☐ 一阶差分
- ☐ 一阶对数差分

ADF平稳性检验结果

ADF统计量

-1.59

p值

0.49不平稳 ($p \geq 0.05$)

正态性检验结果

Shapiro-Wilk检验统计量

0.98

p值

0.00不符合正态分布 ($p < 0.05$)

Ljung-Box白噪声检验结果

Q统计量 (滞后阶数=1)

244.54

p值 (滞后阶数=1)

0.00序列不是白噪声 ($p < 0.05$)

自相关检测结果

ACF函数拖尾

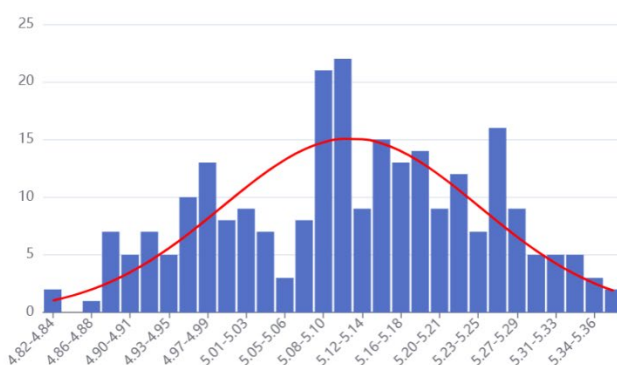
PACF函数2阶截尾

模型建议: 数据可能符合AR(2)模型

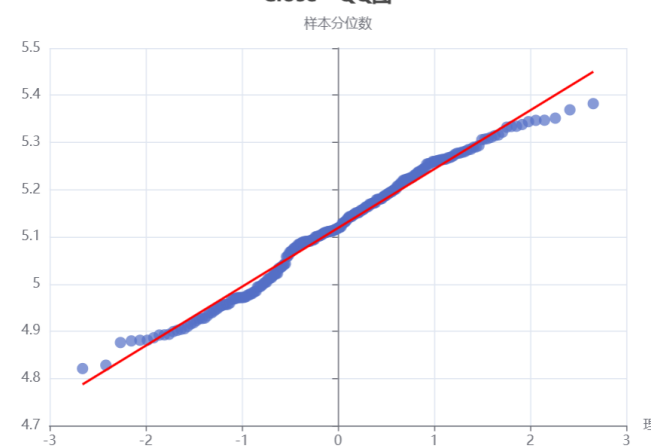
Close - 对数变换后的数据



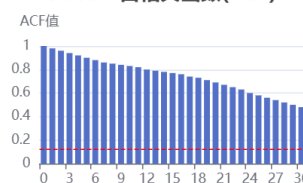
Close - 分布直方图



Close - QQ图



Close - 自相关函数(ACF)



Close - 偏自相关函数(PACF)



图 4.6 ARIMA 模型界面

MSE	RMSE	MAE	方向准确率
0.0185	0.1358	0.1151	0.5714

ARIMA预测结果



模型残差



残差分布

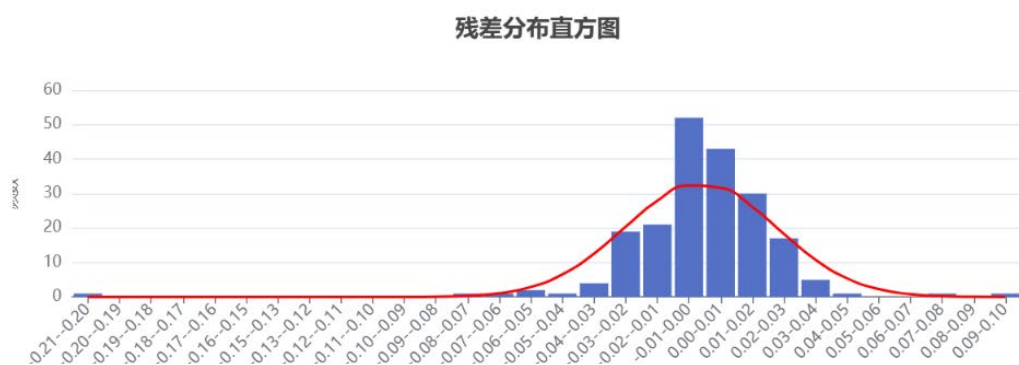


图 4.7 ARIMA 预测结果相关图表

4.4 模型评估模块设计

4.4.1 多模型比较框架

模型评估模块采用标准化的性能比较框架,支持 LSTM 和 ARIMA 模型的横向对比分析。系统首先通过数据对齐算法处理不同模型预测结果的长度差异,确保比较的公平性和准确性。评估指标涵盖精度类指标(MSE, RMSE, MAE, R^2)、相对误差指标(MAPE)和方向性指标(方向准确率),形成多维度的性能评估体系。

可视化比较通过多种图表形式实现,包括性能指标对比表格、多维度雷达图和单指标柱状图。预测结果分析则通过时序对比图和散点图展示,前者直观显示各模型的预测轨迹,后者通过拟合优度(R^2)量化预测精度。

4.4.2 评估报告生成系统

系统设计了灵活的评估报告生成机制,支持 HTML、Markdown 和 JSON 三种输出格式。报告内容采用模块化组织,用户可根据需要选择包含的章节,如执行摘要、性能指标详情、预测分析等。报告生成过程中,系统自动处理数据类型转换问题,确保 NumPy 数组等特殊类型能够正确序列化为标准格式。

4.5 核心工具函数设计

4.5.1 数据处理工具集

数据处理工具集提供了系统运行所需的基础数据操作功能。`normalize_column_names()` 函数实现智能列名标准化, `fix_datetime_for_arrow()` 函数解决时间戳格式兼容性问题。这些工具函数采用纯函数设计模式,确保操作的幂等性和可预测性。

4.5.2 图表生成工具集

图表生成工具集封装了 ECharts 配置的标准化生成逻辑,每个函数负责特定类型图表的配置生成。函数设计遵循单一职责原则,输入标准化的数据结构,输出符合 ECharts 规范的配置字典。这种设计模式不仅提高了代码复用性,也确保了图表样式的一致性。

4.5.3 会话管理工具集

会话管理工具集提供了对 Streamlit 会话状态的标准化访问接口。`get_state()` 和 `set_state()` 函数封装了基本的状态操作, `update_states()` 函数支持批量状态更新。这些接口函数通过统一的错误处理和默认值机制,提高了状态管理的健壮性和可靠性。

通过上述模块化设计,系统实现了功能的有效分离和组件的高度复用,为后续的功能扩展和性能优化奠定了坚实的架构基础。

5 系统实现

本章从技术实现的角度阐述系统核心功能的具体实现方法，包括关键算法的选择、核心库的应用以及关键代码逻辑的设计。系统实现严格遵循前述设计方案，通过模块化编程和标准化接口确保代码的可维护性和可扩展性。

5.1 技术栈与开发环境

系统基于 Python 3.8+ 构建，充分利用其在数据科学和机器学习领域的成熟生态系统。核心技术栈包括 Streamlit 作为 Web 应用框架，Pandas 和 NumPy 提供数据处理和数值计算支持，Scikit-learn 承担机器学习相关任务，Statsmodels 实现统计建模功能，PyTorch 构建深度学习模型，TA-Lib 计算技术指标，以及 Streamlit-Echarts 实现交互式数据可视化。开发环境配置考虑了跨平台兼容性和 GPU 加速需求。系统支持 CPU 和 GPU 两种计算模式，其中 GPU 模式主要用于加速 LSTM 模型的训练过程。通过 requirements.txt 文件管理依赖版本，确保开发和部署环境的一致性。

5.2 数据处理与预处理实现

5.2.1 数据标准化机制

数据标准化通过 `normalize_column_names()` 函数实现，该函数维护了一个完整的列名映射字典，支持中英文混合命名和非标准格式的自动识别与转换。映射规则涵盖了常见的金融数据列名变体，如“日期”映射为“Date”，“收盘价”映射为“Close”等。函数采用字典查找的方式进行列名匹配，确保转换过程的高效性和准确性。

5.2.2 技术指标计算引擎

技术指标计算基于 TA-Lib 库实现，通过 `calculate_technical_indicators()` 函数封装了多种金融技术指标的计算逻辑。系统首先检查输入数据的完整性，确认是否包含必需的 OHLCV 列，然后根据数据可用性选择性计算相应指标。对于数据不完整的情况，系统采用优雅降级策略，提供部分可用指标的计算结果。指标计算涵盖趋势类指标(如移动平均线、MACD)、动量类指标(如 RSI、随机指标)、波动率类指标(如布林带、ATR)和成交量类指标(如 OBV)。每类指标的计算都通过相应的 TA-Lib 指标类实现，如 RSI 通过 `ta.momentum.RSIIndicator(df['Close'], window=14).rsi()` 计算。对于 TA-Lib 无法处理的特殊情况，系统提供了基于 Pandas 滚动窗口的备用计算逻辑。

5.3 LSTM 深度学习模型实现

5.3.1 特征工程与选择

LSTM 模型的特征选择采用多阶段筛选策略，通过 `select_features()` 函数实现。该函数首先计算所有候选特征与目标变量之间的皮尔逊相关系数，保留相关性绝对值超过用户设定阈值的特征。随后采用方差膨胀因子(VIF)检测多重共线性，迭代移除 VIF 值最高的特征直至所有特征的 VIF 值均低于阈值。最后通过 F 检验计算统计显著性，保留 P 值小于设定阈值的特征。这种渐进式筛选策略在保证特征预测能力的同时，有效避免了维度灾难和多重共线性问题。每个筛选阶段的结果都被详细记录，为用户提供透明的特征选择过程。

5.3.2 序列数据构造与归一化

时间序列数据的监督学习转换通过 `create_sequences()` 函数实现。该函数根据用户指定的序列长度和预测长度，采用滑动窗口技术将时间序列数据转换为输入-输出对。具体而言，对于序列长度为 n 的设置，函数提取连续 n 个时间步的数据作为输入特征，对应的下一个时间步数据作为预测目标。数据归一化采用 `MinMaxScaler` 实现，将特征数据和目标数据分别归一化至[0,1]区间。特征数据按列独立归一化，确保不同量纲的特征得到平衡处理。归一化器对象被妥善保存，用于训练完成后的预测结果反归一化处理。

5.3.3 神经网络架构与训练

LSTM 模型通过继承 PyTorch 的 `nn.Module` 类实现，封装了多层 LSTM 网络结构和全连接输出层。网络架构支持灵活的超参数配置，包括隐藏层维度、网络深度、Dropout 正则化率等关键参数。前向传播过程中，数据首先通过 LSTM 层提取时序特征，然后通过全连接层映射到预测目标维度。训练过程采用标准的监督学习范式，使用均方误差作为损失函数，Adam 优化器进行参数更新。系统支持训练集/验证集划分和早停机制，通过监控验证损失防止过拟合。训练过程的损失曲线通过 Streamlit 组件实时展示，为用户提供直观的训练监控体验。

5.4 ARIMA 统计建模实现

5.4.1 时间序列预分析

ARIMA 建模严格遵循 Box-Jenkins 方法论，首先通过一系列统计检验分析时间序列的基本特征。平稳性检验采用增广迪基-富勒(ADF)检验法，通过 `check_stationarity()` 函数封装。白噪声检验采用 Ljung-Box 检验法，通过 `check_white_noise()` 函数实现。自相关和偏自相关分析通过 `analyze_acf_pacf()` 函数计算 ACF 和 PACF 值及其置信区间。这些预分析工具不仅为模型定阶提供统计依据，也帮助用户深入理解数据的时序特征。分析结果通过标准化的图表配置函数生成相应的 ECharts 可视化，为用户提供直观的分析结果。

5.4.2 参数估计与模型拟合

ARIMA 模型的核心拟合通过 Statsmodels 库的 `ARIMA` 类实现，系统通过 `fit_arima_model()` 函数封装了模型拟合过程并处理可能的拟合异常。参数确定支持自动寻优和手动配置两种模式。自动寻优通过 `find_best_arima_params()` 函数实现，采用网格搜索策略在用户指定的参数空间内寻找最优的(p,d,q)组合。优化目标基于 AIC 或 BIC 信息准则，通过遍历所有参数组合并比较相应的信息准则值确定最优参数。手动配置模式则允许有经验的用户直接指定参数，适用于对数据特征有深入了解的应用场景。

5.4.3 模型诊断与残差分析

模型诊断通过 `check_residuals()` 函数实现，对拟合完成的 ARIMA 模型进行充分性检验。该函数从模型结果对象中提取残差序列，通过时序图和分布图展示残差的基本特征，通过 Ljung-Box 检验验证残差的白噪声特性，通过 ACF 和 PACF 图检查残差的自相关性。理想情况下，充分的 ARIMA 模型应产生无自相关的白噪声残差。残差分析结果为用户提供了模型充分性的统计证据，确保模型满足 ARIMA 的基本假设条件。

5.5 模型评估与比较实现

5.5.1 数据对齐与指标计算

多模型比较的关键挑战在于不同模型预测结果的数据对齐。`get_prediction_data()` 函数实现了标准化的数据对齐算法，处理 LSTM 和 ARIMA 模型预测序列长度差异的问题。该函数基于日期索引或序列长度进行对齐，通常采用裁剪到共同最短长度的策略，确保比较的公平性和准确性。评估指标的计算涵盖精度类指标(MSE, RMSE, MAE, R^2)、相对误差指标(MAPE)和方向性指标(方向准确率)。指标计算严格按照统计学定义实现，确保评估结果的准确性和可比性。

5.5.2 可视化与报告生成

系统的可视化功能基于 Apache ECharts 实现，通过 `streamlit-echarts` 库与 Streamlit 无缝集成。每种图表类型都有对应的配置生成函数，如 K 线图配置通过 `create_echarts_kline_volume()` 生成，相关性热力图配置通过 `create_correlation_heatmap()` 生成。这些函数输入标准化的数据结构，输出符合 ECharts 规范的配置字典。

评估报告生成通过 `generate_evaluation_report()` 函数实现，支持 HTML、Markdown 和 JSON 三种输出格式。报告内容采用模块化组织，用户可根据需要选择包含的章节。报告生成过程中，系统通过 `make_json_serializable()` 函数处理 NumPy 数组等特殊数据类型的序列化问题，确保报告的正确生成和导出。

模型状态概览

已训练模型

2个

↑ LSTM + ARIMA

最佳模型

LSTM

↑ MSE: 74.2806

LSTM RMSE

8.6186

↑ 已训练

ARIMA RMSE

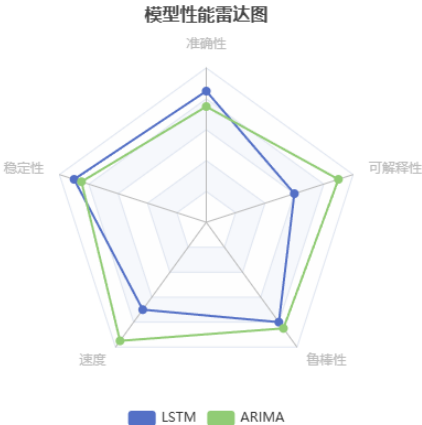
23.8167

↑ 已训练

模型对比 预测分析 详细报告

模型性能对比

性能指标对比



模型	MSE	RMSE	MAE	方向性
LSTM	74.2806	8.6186	7.1548	N/A
ARIMA	567.2360	23.8167	20.0988	N/A

推荐模型: LSTM

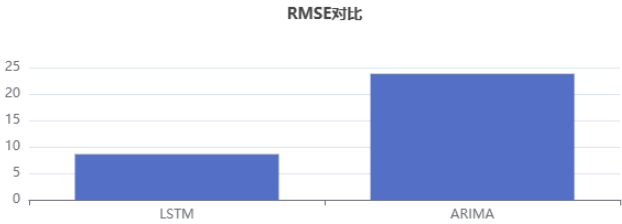
基于MSE指标评选

指标对比图



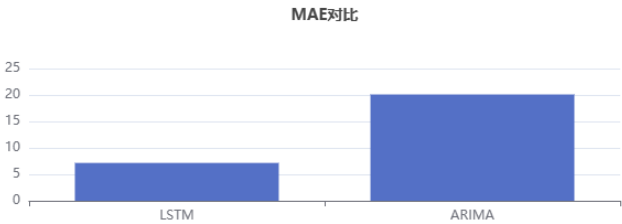
最佳: LSTM

最佳值
74.2806



最佳: LSTM

最佳值
8.6186



最佳: LSTM

最佳值
7.1548

图 5.1 模型对比界面



图 5.2 分析报告查看与导出界面

5.6 会话状态管理实现

系统通过 Streamlit 的 `session_state` 机制实现跨页面的数据持久化和状态同步。为规范化状态管理操作，系统在 `src/utils/session.py` 中封装了标准化的状态访问接口。`get_state()` 函数提供安全的状态获取功能，支持默认值设置和自动初始化。`set_state()` 函数实现状态设置功能，`update_states()` 函数支持批量状态更新。这些接口函数通过统一的错误处理和默认值机制，提高了状态管理的健壮性和可靠性。标准化的状态管理接口确保了系统各模块间数据传递的一致性和可预测性，为系统的稳定运行提供了重要保障。

通过上述实现方案，系统成功将理论设计转化为可运行的软件系统，为用户提供了完整的时间序列预测分析平台。各模块的实现都严格遵循软件工程的最佳实践，确保了系统的可维护性、可扩展性和稳定性。

6 系统测试与分析

本章从软件质量保证的角度阐述系统测试的设计思路、实施过程和结果分析。测试工作遵循软件工程的标准化流程，通过功能验证、性能评估和缺陷分析，确保系统的可靠性、稳定性和用户体验质量。

6.1 测试方法论与环境配置

6.1.1 测试策略设计

系统测试采用多层次验证策略，结合手动测试和探索性测试方法。针对核心算法和数据处理逻辑，采用单元测试方法进行精确验证；对于用户交互流程，通过模拟真实使用场景进行端到端测试。这种分层测试策略既保证了算法实现的正确性，又验证了系统整体功能的完整性。测试过程严格遵循迭代开发模式，每个功能模块完成后立即进行相应测试，确保问题能够及时发现和修复。同时，建立了回归测试机制，在系统修改后重新验证已有功能的稳定性。

6.1.2 测试环境构建

测试环境的配置考虑了系统的跨平台特性和多样化的硬件需求。主要测试平台为 Windows 操作系统，浏览器环境以 Chrome 为主，同时兼顾其他主流浏览器的兼容性。硬件配置涵盖了 CPU 和 GPU 两种计算模式，特别针对 LSTM 模型的 GPU 加速功能进行了专门测试。Python 环境管理通过虚拟环境实现，确保依赖库版本的一致性和测试结果的可重现性。测试数据集包括多种规模和特征的时间序列数据，以验证系统在不同数据条件下的表现。

6.2 功能验证与性能评估

6.2.1 数据处理模块验证

数据处理模块的测试重点关注数据标准化、技术指标计算和可视化功能的准确性。通过构造包含不同列名格式、数据规模和完整性的测试数据集，验证系统的数据适应能力和容错机制。技术指标计算的准确性通过与标准金融软件的结果对比进行验证，确保计算逻辑的正确性。可视化功能的测试涵盖了 K 线图、相关性热力图等多种图表类型，重点验证图表渲染的正确性、交互功能的有效性以及数据导出的完整性。测试结果表明，系统能够正确处理各种数据格式，并在数据不完整的情况下提供合理的降级方案。

6.2.2 模型训练与评估验证

LSTM 模型训练模块的测试涵盖了特征选择、超参数配置、训练过程监控和结果展示等关键环节。特征选择算法的有效性通过统计显著性检验进行验证，确保筛选出的特征具有良好的预测能力。训练过程的稳定性通过多次重复实验进行评估，验证早停机制和损失函数收

敛的正确性。ARIMA 模型测试重点关注时间序列预分析、参数自动寻优和残差诊断功能。平稳性检验、白噪声检验等统计测试的实现通过标准统计软件的结果进行对比验证。参数自动寻优功能在多个数据集上进行了测试，验证其能够找到统计意义上的最优参数组合。模型评估模块的测试重点验证多模型比较的公平性和准确性。数据对齐算法通过构造不同长度的预测序列进行测试，确保比较结果的可靠性。评估指标的计算精度通过手工计算进行验证，确保统计指标的正确性。

6.3 系统性能分析

6.3.1 计算性能评估

系统的计算性能评估涵盖了数据处理、模型训练和结果生成等关键环节。对于中等规模的时间序列数据（数万行级别），数据加载和技术指标计算能够在秒级时间内完成，满足实时分析的需求。LSTM 模型的训练性能受多种因素影响，包括数据集规模、特征维度、网络复杂度和硬件配置等。在 GPU 加速环境下，训练效率相比 CPU 模式有显著提升。ARIMA 模型的单次拟合速度较快，但参数自动寻优过程的时间复杂度随参数空间的增大而显著增加。

6.3.2 用户体验评估

用户体验评估主要通过界面交互测试和操作流程分析进行。Streamlit 框架提供的组件库能够满足系统的交互需求，页面响应速度和导航逻辑得到了用户的积极反馈。参数配置界面的设计兼顾了专业性和易用性，通过提示信息和默认值设置降低了使用门槛。错误处理机制的测试表明，系统能够对常见的用户操作错误提供明确的提示信息，并引导用户进行正确操作。数据导出和报告生成功能的可靠性通过多种格式和内容组合进行了验证。

6.4 缺陷识别与解决方案

1. 关键技术问题分析

在系统开发和测试过程中，识别并解决了多个关键技术问题。图表渲染问题主要源于数据格式不匹配和函数调用错误，通过标准化数据处理流程和统一图表生成接口得到解决。模块导入路径问题通过调整项目结构和优化导入机制得到改善。数据类型兼容性问题在 Streamlit 与第三方库集成时较为常见，特别是时间戳格式和 NumPy 数组的序列化处理。通过实现专门的数据类型转换函数，系统能够处理各种数据格式的兼容性问题。

2. 系统稳定性改进

多模型预测结果的数据对齐是系统设计中的重要挑战。不同模型的预测序列在长度和时间对应关系上可能存在差异，影响比较结果的准确性。通过设计标准化的数据对齐算法和用户可选的对齐策略，系统能够确保模型比较的公平性和可靠性。会话状态管理的稳定性通过封装标准化接口得到保障。统一的状态访问机制减少了因状态管理不当导致的系统错误，提高了跨页面数据传递的可靠性。

7 总结与展望

本研究成功设计并实现了基于 Streamlit 的多模型时间序列预测系统，该系统集成了传统计量方法与深度学习技术，为金融时间序列分析提供了完整的建模 workflow。通过系统化的需求分析、架构设计、功能实现和测试验证，项目达到了预期的研究目标，为时间序列预测领域的理论研究与实际应用搭建了有效的桥梁。

7.1 研究成果总结

1. 系统功能实现。本系统实现了涵盖数据预处理、特征工程、模型训练和性能评估的完整分析流程。数据处理模块通过智能列名标准化和多维度技术指标计算，为后续建模提供了丰富的特征变量。LSTM 深度学习模块采用多阶段特征筛选策略，结合灵活的网络架构配置，实现了对复杂非线性时序模式的有效捕捉。ARIMA 统计建模模块严格遵循 Box-Jenkins 方法论，通过完整的时序分析工具链，确保了模型的统计有效性。模型评估框架建立了多维度的性能比较体系，通过标准化的数据对齐算法和综合评估指标，实现了不同建模范式间的公平比较。交互式可视化系统基于 Apache ECharts 构建，为用户提供了直观的数据探索和结果分析体验。

2. 技术创新与贡献。系统的主要技术贡献体现在多个方面。首先，通过集成传统统计方法与现代深度学习技术，为用户提供了多样化的建模选择，充分发挥了不同方法的各自优势。其次，基于 Streamlit 框架的 Web 应用设计显著降低了高级时序分析技术的使用门槛，使得复杂的建模过程变得直观可操作。特征工程方面的创新在于实现了基于统计显著性的自动化特征筛选机制，有效解决了高维特征空间中的维度灾难和多重共线性问题。模型评估体系的设计兼顾了统计精度和金融应用的实际需求，通过方向准确率等专门指标，更好地反映了模型在实际投资决策中的价值。交互式可视化系统基于 Apache ECharts 构建，为用户提供了直观的数据探索和结果分析体验。

3. 实际应用价值。系统具有重要的教育和研究价值。对于金融计量学习者而言，系统提供了理论与实践相结合的学习平台，通过可视化的界面和实时反馈机制，帮助用户深入理解不同建模方法的工作原理和适用场景。对于研究人员而言，系统的模块化设计和标准化接口为新方法的集成和比较提供了便利的实验环境。在实际应用层面，系统为金融机构的量化分析团队提供了快速原型开发和模型验证的工具。通过自动化的报告生成功能，分析结果可以方便地与决策层分享，提高了研究成果的传播效率。

7.2 系统局限性分析

1. 技术层面的限制。当前系统在技术实现上存在一定局限性。计算性能方面，虽然系统支持 GPU 加速，但对于大规模数据集的处理能力仍有待提升。模型种类方面，目前仅集成了 LSTM 和 ARIMA 两种代表性方法，尚未涵盖 Transformer、Prophet 等新兴的时序预测技术。

特征工程的自动化程度有限，主要依赖传统的技术指标计算，缺乏更高级的特征生成和选择机制。模型解释性方面，特别是对于 LSTM 等深度学习模型，系统尚未提供充分的可解释性分析工具。

2. 应用场景的约束。系统的应用范围主要局限于单变量时间序列预测，对于多变量时序建模和复杂的宏观经济因子分析支持不足。数据源方面，目前仅支持 CSV 格式的静态数据导入，缺乏实时数据流处理和多源数据融合能力。

3. 金融应用的专业性。在金融应用的专业性方面，系统缺乏风险管理、投资组合优化等高级金融建模功能，限制了其在实际投资决策中的直接应用价值。

7.3 发展方向与改进策略

1. 技术架构优化。未来的技术发展将重点关注系统的可扩展性和性能优化。通过引入微服务架构和容器化部署，提高系统的可维护性和部署灵活性。计算性能方面，考虑集成分布式计算框架，支持大规模数据的并行处理。缓存机制的优化将显著改善用户体验，特别是对于重复计算任务的处理效率。模型库的扩展是技术发展的重要方向。计划集成 Transformer、Prophet、N-BEATS 等先进的时序预测模型，为用户提供更丰富的建模选择。同时，引入自动机器学习(AutoML)技术，实现超参数的智能优化和模型的自动选择。

2. 功能模块增强。特征工程模块的增强将重点关注自动化特征生成和高级特征变换技术。通过引入小波变换、傅里叶分析等信号处理方法，提升系统对复杂时序模式的识别能力。特征选择算法的改进将结合最新的机器学习理论，提供更精确的特征重要性评估。模型解释性的提升是功能发展的重要方向。计划集成 SHAP、LIME 等可解释性分析工具，为深度学习模型提供透明的决策解释。这对于金融应用中的监管合规和风险控制具有重要意义。

3. 应用领域拓展。系统的应用领域拓展将重点关注多变量时序建模和宏观经济分析。通过引入向量自回归(VAR)、动态因子模型等多变量时序方法，扩展系统的分析能力。同时，考虑集成经济指标数据源，为宏观经济预测提供专门的建模工具。

4. 实时数据处理能力的建设将使系统能够应对高频交易和实时风险监控的需求。通过集成流数据处理框架，实现对市场数据的实时分析和预警。

7.4 学术贡献与实践意义

1. 理论贡献。本研究在时间序列预测的方法论整合方面做出了有益探索。通过系统化地比较传统统计方法与深度学习技术的优劣，为不同应用场景下的方法选择提供了实证依据。特征工程的多阶段筛选策略为高维时序数据的处理提供了有效的解决方案。评估指标体系的设计兼顾了统计意义和实际应用价值，特别是方向准确率等金融专用指标的引入，丰富了时序预测模型的评估维度。

2. 实践价值。系统的实践价值主要体现在降低了高级时序分析技术的应用门槛。通过用户友好的界面设计和自动化的分析流程，使得复杂的建模技术能够被更广泛的用户群体所掌

握和应用。在教育领域，系统为金融计量和时序分析课程提供了有效的实验平台。学生可以通过实际操作加深对理论知识的理解，提高实践能力。在研究领域，系统的标准化接口和模块化设计为新方法的验证和比较提供了便利的实验环境。

7.5 结语

本研究通过系统化的设计和实现，成功构建了一个功能完善的多模型时间序列预测平台。虽然当前系统在某些方面仍存在局限性，但其在方法整合、技术实现和应用推广方面的贡献为时间序列预测领域的发展提供了有价值的参考。

随着人工智能和金融科技的快速发展，时间序列预测技术将在更多领域发挥重要作用。本系统的设计理念和技术架构为未来的发展奠定了坚实基础。通过持续的技术创新和功能完善，该系统有望成为时间序列分析领域的重要工具，为学术研究和实际应用提供更强有力的支持。

参考文献

- [1] Guerard J B, Markowitz H, Xu G. Earnings forecasting in a global stock selection model and efficient portfolio construction and management[J]. International Journal of Forecasting, 2015, 31(2): 550-560.
- [2] Paiva F D, Cardoso R T N, Hanaoka G P, et al. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection[J]. Expert Systems with Applications, 2019, 115: 635-655.
- [3] 刘楠. 多因子投资模型的因子拓展: 行为金融学角度[J]. 中国外资, 2022(12): 108-111.
- [4] 闫宇, 吴海涛. 基于 arima 模型的纳斯达克指数短期预测[J]. 信息与电脑(理论版), 2020, 32(20): 155-158.
- [5] 吴鑫育, 赵安, 谢海滨, 等. 中国股市波动率预测研究: 基于实时已实现 egarch-midas 模型[J]. 计量经济学报, 2024, 4(1): 248-273.
- [6] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. 2023.
- [7] 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.
- [8] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Burstein J, Doran C, Solorio T. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186.
- [9] Darapaneni N, Paduri A R, Sharma H, et al. Stock price prediction using sentiment analysis and deep learning for Indian markets[A]. arXiv, 2022.
- [10] Si W, Li J, Ding P, et al. A multi-objective deep reinforcement learning approach for stock index future's intraday trading[C]//2017 10th International Symposium on Computational Intelligence and Design (ISCID): Vol. 2. 2017: 431-436.
- [11] 王钧弘. 初探深度学习模型在国债 ETF 中的预测应用——基于 CNN-LSTM-attention 混合架构[J]. 债券, 2025(3): 15-21.
- [12] Maiti A, Shetty D P. Indian stock market prediction using deep learning[C]//2020 IEEE REGION 10 CONFERENCE (TENCON). 2020: 1215-1220.
- [13] 张伟豪. 基于长短期记忆神经网络的股票时间序列预测[J]. 信息与电脑(理论版), 2022, 34(9): 68-72.
- [14] Lu W, Li J, Wang J, et al. A CNN-BiLSTM-AM method for stock price prediction[J]. Neural Computing and Applications, 2021, 33(10): 4741-4753.

- [15] 刘祺, 施三支, 娄磊, 等. 基于 ariam-garch 深度学习的股价预测与决策[J]. 长春理工大学学报 (自然科学版), 2024, 47(1): 119-130.
- [16] 危冰淋, 刘春雨, 刘家鹏. 基于 Transformer-LSTM 模型的多因素碳排放权交易价格预测[J]. 价格月刊, 2024(5): 49-57.
- [17] 夏雪, 闫恩来, 李喜武. Transformer 在时间序列预测中的应用综述[J]. 信息技术与信息化, 2024(3): 124-128.
- [18] Jiang J, Han C, Zhao W X, et al. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(4): 4365-4373.
- [19] 张虎, 沈寒蕾, 刘晔诚. 基于自注意力神经网络的多因子量化选股问题研究[J]. 数理统计与管理, 2020, 39(3): 556-570.
- [20] Fama E F. Efficient capital markets: a review of theory and empirical work[J]. The Journal of Finance, 1970, 25(2): 383.
- [21] Engle R F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation[J]. Econometrica, 1982, 50(4): 987-1007.
- [22] 闫洪举. 基于深度学习的指数跟踪方法研究[J]. 统计与决策, 2021, 37(5): 143-147.
- [23] 苏治, 卢曼, 李德轩. 深度学习的金融实证应用: 动态、贡献与展望[J]. 金融研究, 2017(5): 111-126.
- [24] Hajiabotorabi Z, Kazemi A, Samavati F F, et al. Improving DWT-RNN model via B-spline wavelet multiresolution to forecast a high-frequency time series[J]. Expert Systems with Applications, 2019, 138: 112842.
- [25] Zhou X, Pan Z, Hu G, et al. Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets[J]. Mathematical Problems in Engineering, 2018, 2018: 1-11.
- [26] Darapaneni N, Paduri A R, Sharma H, et al. Stock price prediction using sentiment analysis and deep learning for Indian markets[A]. arXiv, 2022.
- [27] Shi Z, Hu Y, Mo G, et al. Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction[A]. arXiv, 2022.