

基于量化投资问题的模型建立与研究

摘要

本文主要研究有关“数字经济”板块量化投资问题，建立起相关性模型来对与“数字经济”板块有关的主要指标进行提取，并基于此提出了关于“数字经济”板块指数每 5 分钟成交量与每 5 分钟的收盘价的时间序列预测模型。然后在给出了量化交易中的各项策略，最终在不同情况下选择不同的交易策略来计算出特定期间交易的总收益率、信息比率、最大回撤率。

针对问题一：我们首先基于附件所给的市场各项指数数据进行筛选和清洗，然后构建出典型相关性模型，利用 **SPSS** 软件对所提供的各项指标与“数字经济”板块间的数据进行分析。通过分析结果的典型相关系数、标准化典型相关系数，并进行 F 值检验，我们最终得出了与“数字经济”板块有关的 8 个指标，详情见附录 11.2.1。

针对问题二：基于问题一中所提出的 8 项主要指标，我们首先利用 **Python** 中的 **pandas** 库对“数字经济”板块指数数据集进行数据补充。之后本文提出了 **ARIMA** 时间序列预测模型来对“数字经济”板块指数每 5 分钟成交量进行预测。但可能是数据并没有明显的时间序列性，导致模型的预测误差均大于 10^5 。我们再次根据数据的无明显的时间序列性这一特点提出 **BP 神经网络预测模型**，求解结果表明，模型的相关系数为 0.7609，具有较好的拟合程度和预测准确率，最终通过求解模型确定“数字经济”板块指数每 5 分钟成交量，详情见附录 11.2.2

针对问题三：考虑到问题二中神经网络对预测结果取得良好效果，我们提出了更加适合时间序列预测的循环神经网络——**LSTM** 和 **GRU** 预测模型来对每 5 分钟的“数字经济”板块指数（收盘价）进行预测。求解结果表明，**LSTM** 与 **GRU** 预测模型的 **RSE** 分别为 **2.991437**、**3.607073**。最后，我们通过选择求解 **LSTM** 模型确定“数字经济”板块指数每 5 分钟的收盘价，详情见附录 11.2.3。

针对问题四：通过查阅相关参考文献和股市实际情况，我们首先提出了四种不同的交易策略。并通过取对数收益率计算出 **Hurst** 值，根据 **Hurst** 值的大小从上述四种决策中来选择不同的策略，最终计算出 2022 年 1 月 4 日至 2022 年 1 月 28 日期间交易的总收益率、信息比率、最大回撤率分别为：**8.24%**、**45%**、**9.9%**。

关键字：典型相关性分析 **BP 神经网络** **LSTM** **GRU** **Python**

目录

一、问题重述	3
二、问题分析	3
三、模型的假设	4
四、符号说明	4
五、问题一的模型建立与求解	4
5.1 典型相关分析模型的建立	5
5.1.1 “数字经济”板块与技术指标典型相关分析结果	6
5.1.2 “数字经济”板块与国内市场指标典型相关分析结果	7
5.1.3 “数字经济”板块与国际市场指标典型相关分析结果	9
5.1.4 “数字经济”板块与其他板块指标典型相关分析结果	10
六、问题二的模型建立与求解	10
6.1 数据的预处理	10
6.2 问题二的模型建立	11
6.2.1 基于 ARIMA 的每 5 分钟成交量预测模型	11
6.2.2 基于 BP 神经网络的每 5 分钟成交量预测模型	13
七、问题三的模型建立与求解	15
7.1 数据的处理与预处理	15
7.2 基于 LSTM 的收盘价预测模型	15
八、问题四的模型建立与求解	17
8.1 数据的预处理	17
8.2 交易策略的制定	18
8.3 交易模型的建立与求解	19
九、模型分析	19
9.1 模型优点	19
9.2 模型缺点	20
9.3 模型的改进	20
十、参考文献	20
十一附录	20

一、问题重述

为获取稳定收益，通过数量化方式及计算机程序化发出买卖指令的交易方式称为量化投资。合理进行量化投资进而完成交易，需要了解市场运行规律，预测市场走势，但庞大的市场信息以及受多种因素影响而波动的商品价格给提取有效指标进而指定交易策略增加了难度。探索如何建立正确指标，依靠计算机技术，高效开展投资决策是一项有意义的课题。

基于附件给出的 2021 年 7 月 14 日至 2022 年 1 月 28 日每 5 分钟的“数字经济”板块（以下简称：板块）给出的数据信息，我们团队逐一完成了以下问题：

1. 对所提供的各项指标进行分析，从中提取出与“数字经济”板块有关的主要指标。
2. 以 2021 年 7 月 14 日至 2021 年 12 月 31 日的板块指数为训练集，以 2022 年 1 月 4 日至 2022 年 1 月 28 日的板块指数为测试集。根据问题 1 提取出来的各项指标对“数字经济”板块指数每 5 分钟成交量进行预测。
3. 以 2021 年 7 月 14 日至 2021 年 12 月 31 日的板块指数为训练集，以 2022 年 1 月 4 日至 2022 年 1 月 28 日的板块指数为测试集。根据 1 和 2 建立模型对每 5 分钟的“数字经济”板块指数（收盘价）进行预测。
4. 假设以板块指数为交易对象（在实际交易中指数无法交易，只能交易其中的个股），给定初始资金 100 万元，交易佣金为 0.3%，根据（3）得到的结果对板块每 5 分钟频率价格进行买卖交易，计算在 2022 年 1 月 4 日至 2022 年 1 月 28 日期间交易的总收益率、信息比率、最大回撤率。

二、问题分析

- 针对问题一：

问题一要求对所提供的各项指标进行分析最终提取出与“数字经济”板块有关的主要指标。由于涉及到多个变量之间的关系，我们将他们分成两组，采用典型相关分析法，建立模型，对指标之间的相关性进行分析以确定主要指标。

- 针对问题二：

问题二要求我们以“数字经济”板块给出的数据，将其分为训练集和测试集，根据问题一得出的指标对每五分钟的成交量进行预测。由于数据单位跨度的不同，我们首先可以对数据进行清洗，得到每五分钟的对应数据，接着再通过两种建立的模型进行预测，挑选效果较好的那一种作为结果。

- 针对问题三：

问题三同样要求我们以“数字经济”板块给出的数据，将其分为训练集和测试集，根据问题一、问题二建立模型，对收盘价进行预测。我们可以利用已获得的有关数据集，以时间作为数据的索引项，建立不同的预测模型。考虑到数据可能空缺，所以我们需要对空白纸进行切除和拼接。通过比较两种预测模型的结果，最后确定

选取哪一个模型进行预测。

• 针对问题四：

问题四要求我们假设以“数字经济”板块指数为交易对象，以给定的初始资金和交易佣金，根据（3）得到的结果对“数字经济”板块每 5 分钟频率价格进行买卖交易，计算在给定时间内交易的总收益率、信息比率、最大回撤率。可以由 Hurst 值上分别对几种不同策略进行分析。

三、模型的假设

- 1. 假设我们所利用的指标数据都是准确、可靠和互相一致的。
- 2. 假设我们在模型的提出和验证过程中，我们忽略的其他市场指标对结果的计算影响不大。
- 3. 假设股市市场是长期稳定的，并不会出现股价剧烈波动情况。
- 4. 假设进行买卖交易是瞬时的，不会有时间条件约束。

四、符号说明

符号	意义
S_{11} 和 S_{12}	“数字经济”板块和国内市场指标内的相关系数阵
S_{21} 和 S_{22}	“数字经济”板块和国际市场指标间的相关系数阵
R	典型相关系数
AR	一个回归模型
$I(d)$	取差 d 次，直到原始序列变为平稳
MA	移动平均模型对过去的预测误差使用类似回归的模型

五、问题一的模型建立与求解

问题一要求通过对各项指标进行分析来提取出与“数字经济”板块有关的主要指标。这个问题对数字经济和市场其他数据指标两组变量的关系分析提出了要求，对此本文通过建立典型相关分析模型进行指标提取。

5.1 典型相关分析模型的建立

由于题中所给指标数量偏多，为了具体研究“数字经济”板块与其他市场指标之间的相关性，我们分别以“数字经济”板块和其他指标作为输入变量与输出变量，采用典型相关分析法（CCA）。CCA 充分利用主成分思想，从两组变量中提取出一个或少数几个综合变量（即典型变量），从而使得少数几对综合变量的关系能较好地代表两组变量之间的关系。

以“数字经济”板块与国内市场指标为例，典型相关分析模型建立的具体步骤如下：

Step1. 原始矩阵的建立

根据附录中所给的“数字经济”板块信息，我们首先对数据进行预处理，除去板块中“时间”和“指数代码”这两项。接着设“数字经济”板块矩阵为 $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_6)'$ ，国内市场指标项矩阵为 $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_{12})'$

Step2. 数据标准化变换及相关系数矩阵的计算

对“数字经济”板块和国内市场指标项矩阵数据进行标准化处理，然后计算两样本间的相关系数矩阵 \mathbf{S} ：

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

其中， \mathbf{S}_{11} 、 \mathbf{S}_{12} 分别为“数字经济”板块和国内市场指标内的相关系数阵， \mathbf{S}_{21} 、 \mathbf{S}_{22} 为“数字经济”板块和国内市场指标间的相关系数阵。

$$S_{11} = \text{Cov}(A), S_{22} = \text{Cov}(B), S_{12} = S_{12}^T = \text{Cov}(A, B)$$

Step3. 典型相关系数及典型变量的求解

将 \mathbf{A} \mathbf{B} 矩阵中的变量分别组合成两个变量 U 、 V （称为典型相关变量），则用线性表示：

$$\begin{aligned} U_k &= t_k^T A = t_{1k}a_1 + t_{2k}a_2 + \dots + t_{nk}a_n \\ V_k &= h_k^T B = h_{1k}b_1 + h_{2k}b_2 + \dots + h_{mk}b_m \end{aligned}$$

典型相关系数为：

$$R_{(U,V)} = \frac{\text{Cov}(U,V)}{\sqrt{\text{Var}[U] \text{Var}[V]}} = \text{Cov}(U,V) = t_k^T \text{Cov}(A,B)h_k = t_k^T \sum_{12} h_k$$

此时，满足约束条件：

$$\begin{aligned} \text{Var}(U_k) &= \text{Var}(t_k^T A) = t_k^T \Sigma_{11} t_k = 1 \\ \text{Var}(V_k) &= \text{Var}(h_k^T B) = h_k^T \Sigma_{22} h_k = 1 \\ \text{Cov}(U_k, U_i) &= \text{Cov}(U_k, V_i) = \text{Cov}(V_i, U_k) = \text{Cov}(V_k, V_i) = 0 (1 \leq i < k) \end{aligned}$$

Step4. 检验各典型相关系数的显著性

对典型相关系数进行显著性检验。在作两组变量“数字经济”板块 \mathbf{A} 和国内市场指标 \mathbf{B} 的典型相关分析之前，首先应检验两组变量是否相关；如果不相关，即 $\text{Cov}(A, B) = 0$ ，则表明讨论的内容并不具有任何研究意义。

同样的，对于技术指标、国际市场指标等附件中提供的市场数据指标，也通过相同的方法与“数字经济”板块进行典型相关分析，这里我们便不再赘述。但考虑到宏观市场指标与汇率这些指标所提供的数据量过少，我们并未对其进行典型相关性分析而是将其直接舍弃。

5.1.1 “数字经济”板块与技术指标典型相关分析结果

利用 SPSS 软件对“数字经济”板块与技术指标进行了典型相关分析，结果如下：

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.976	19.993	0.015	8.932	84.000	664.072	0.000
2	0.629	0.653	0.310	2.447	65.000	566.316	0.000
3	0.511	0.353	0.513	1.828	48.000	464.291	0.001
4	0.460	0.268	0.694	1.426	33.000	357.192	0.065
5	0.306	0.103	0.881	0.801	20.000	244.000	0.711
6	0.169	0.029	0.972	0.400	9.000	123.000	0.933

表 1 典型相关性结果一

从表中可以看出，第一个典型相关系数较高，表明相应典型变量之间密切相关。且显著性为 0.000（ $P<0.01$ ），通过了 F 值检验，表明在 99% 的置信水平下（ $P<0.01$ ），相应典型变量之间相关关系显著。

变量	1	2	3	4	5	6
VMA	-0.645	-1.274	-1.057	-1.441	2.967	0.551
VMACD	0.302	1.750	0.430	1.418	-2.512	-0.336
ARBR	0.096	0.836	1.208	0.160	-0.192	-0.120
OBV	-0.083	0.206	0.452	-1.231	1.697	1.748
BBI	-3.003	-0.881	-5.993	-4.949	-16.524	-2.841
DMA	-0.152	-0.008	3.107	6.781	9.296	-1.466
MA	0.264	-1.854	2.368	-5.315	7.563	-4.517
EXPMA	-0.600	10.004	9.683	19.778	6.676	6.163
MTM	0.067	0.151	-0.825	1.155	0.472	-0.219
MACD	1.859	-4.564	-8.626	-11.754	-10.306	1.647
BIAS	0.107	-1.735	-0.320	0.205	1.612	-0.764
KDJ	0.079	-0.186	0.820	0.552	.766	0.070
RSI	-0.146	-0.257	0.426	-2.005	-2.153	0.201
BOLL	2.269	-5.253	-2.989	-5.837	.142	-0.492

表 2 标准化典型相关系数结果一

标准化典型相关系数绝对值越大表明该项与典型变量之间的相关关系越强，从表中可以看出典型变量于 BBI 相关系数绝对值最大为 3.003，与 BOLL 的相关系数绝对值次之为 2.269, 从表中可以看出典型变量更多地提取典型变量 **BBI** 与 **BOLL** 的信息。

5.1.2 “数字经济”板块与国内市场指标典型相关分析结果

对“数字经济”板块与国内市场指标进行了典型相关分析的结果如下：

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.979	23.084	0.022	19.012	72.000	1338.749	0.000
2	0.536	0.403	0.522	3.133	55.000	1142.265	0.000
3	0.386	0.175	0.732	2.009	40.000	938.451	0.000
4	0.323	0.117	0.860	1.418	27.000	724.930	0.079
5	0.166	0.028	0.961	0.630	16.000	498.000	0.860
6	0.110	0.012	0.988	0.441	7.000	250.000	0.876

表 3 典型相关性结果二

观察分析可知，第一个典型相关系数较高且显著性为 0.000，表明相应典型变量之间密切相关且相关关系显著。

变量	1	2	3	4	5	6
成交量上证综合指数	-0.014	0.483	1.947	2.151	-0.009	0.172
成交金额上证综合指数	0.073	0.009	-2.315	-1.687	0.399	-0.582
沪市股票流通市值	-0.616	-9.765	-4.947	4.969	1.457	0.760
深市股票流通市值	2.654	7.478	5.880	-8.805	-4.266	-0.087
沪深 300 指数	-0.830	4.492	3.207	-14.403	-7.783	1.639
上证综合指数	13.440	439.938	-171.243	113.539	230.306	-44.997
中证 500 指数	-0.550	-3.867	-1.393	2.238	-2.121	-5.335
创业板指数	0.294	2.153	0.578	-0.710	-0.844	-3.711
上证 50 指数	0.430	-2.917	-5.508	11.441	0.864	-7.115
上证 A 股指数	-12.782	-433.693	174.735	-117.077	-226.879	46.935
深证成份指数	0.309	-6.833	0.940	7.616	6.693	3.308
深证综合指数	-1.761	1.359	-4.370	2.826	-3.146	1.484

表 4 标准化典型相关系数结果二

从表中可以看出典型变量更多地提取上证综合指数和上证 A 股指数这两项的信息，两项的相关系数分别为 13.440 与-12.78

5.1.3 “数字经济” 板块与国际市场指标典型相关分析结果

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.965	13.712	0.034	15.955	66.000	1187.993	0.000
2	0.548	0.430	0.493	3.408	50.000	1015.840	0.000
3	0.422	0.217	0.705	2.276	36.000	837.422	0.000
4	0.304	0.102	0.858	1.472	24.000	650.269	0.069
5	0.222	0.052	0.945	0.922	14.000	450.000	0.535
6	0.079	0.006	0.994	0.235	6.000	226.000	0.965

表 5 典型相关性结果三

由表中数据可知，第一个典型相关系数较高为 0.965，显著性为 0.000，因此表明在 99% 的置信水平下 ($P<0.01$)，相应典型变量之间密切相关且相关关系显著。

变量	1	2	3	4	5	6
道琼斯工业平均指数	-0.952	0.953	-0.944	0.135	-2.934	-0.888
纳斯达克综合指数	0.185	-0.708	0.406	2.301	-3.390	-0.958
标准普尔 500 指数	0.955	-0.958	0.806	-2.805	9.452	-0.274
美国证交所	0.620	1.378	0.657	-0.546	-2.447	-1.109
恒生指数	-0.187	-1.258	0.254	-0.578	0.164	-1.237
东京日经 225 指数	-0.157	-0.103	-0.336	-0.643	0.389	0.820
伦敦金融时报 100 指数	-0.040	-1.534	-0.969	0.708	-1.613	1.691
法国巴黎 CAC40 指数	-0.778	2.291	-2.044	-1.340	1.442	-0.139
荷兰 AEX 指数	0.245	-0.102	2.183	-3.325	-3.318	-0.029
俄罗斯 RTS 指数	-0.305	0.414	-0.572	0.871	0.968	0.795
意大利 MIB 指数	0.709	-2.830	0.235	3.843	1.491	0.037

表 6 标准化典型相关系数结果三

从中我们提取出**道琼斯工业平均指数**与**标准普尔 500 指数**这两个指标，两项的相关

系数分别为-0.952 与 0.955。

5.1.4 “数字经济”板块与其他板块指标典型相关分析结果

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.962	12.277	0.062	31.615	30.000	942.000	0.000
2	0.354	0.143	0.819	2.438	20.000	783.673	0.000
3	0.184	0.035	0.936	1.325	12.000	627.335	0.199
4	0.176	0.032	0.969	1.274	6.000	476.000	0.268
5	0.020	0.000	1.000				

表 7 典型相关性结果四

观察分析可知，第一个典型相关系数较高为 0.962 且显著性为 0.000，表明在 99% 的置信水平下 ($P<0.01$)，相应典型变量之间密切相关且相关关系显著。

变量	1	2	3	4	5
数字媒体	-0.589	1.194	0.466	0.630	1.742
数字孪生	-0.085	-0.022	1.368	0.899	2.301
快手概念	0.215	-0.210	-0.135	0.626	-1.730
互联网电商	-0.482	-1.255	0.326	0.529	-0.034
互联网	0.426	-0.386	-1.416	-0.465	-0.272

表 8 标准化典型相关系数结果四

从中我们提取出**数字媒体**与**互联网电商**这两个指标，两项的相关系数分别为-0.589 与-0.482。

六、问题二的模型建立与求解

6.1 数据的预处理

在问题一中，我们通过建立典型相关性分析模型得出了与“数字经济”板块有关的 8 个主要指标（详情见附录 11.1.1）。但考虑到附件中 8 个主要指标给出的均为当日的数

据，我们 Python 中的 Pandas 库对所给主要数据与指标进行清洗切割与拼接，获得了包含所有与“数字经济”板块有关指标的数据集。（详情见附录 11.1.2）

6.2 问题二的模型建立

6.2.1 基于 ARIMA 的每 5 分钟成交量预测模型

ARIMA 是一种用于分析和构建预测模型的统计方法，该模型通过对数据中的相关性进行建模来最好地表示时间序列。由于属于纯统计方法，ARIMA 模型只需要时间序列的历史数据就可以进行数据的预测并设法提高预测准确性，同时保持模型简洁性。

ARIMA 模型包含 3 个部分：即自回归（AR）、差分（I）和移动平均（MA）

- **AutoRegressive/AR(p)**：是一个回归模型，其中 y 的滞后值，经过 p 次阶数作为预测变量。这里， p 是模型中滞后观察的数量， ϵ 是时间 t 的白噪声， c 是常数， ϕ_s 是参数。

$$\hat{y}_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t$$

- **差分 I(d)**：取差 d 次，直到原始序列变为平稳。平稳时间序列是其属性不依赖于观察该序列的时间的序列。

$$y'_t = y_t - y_{t-1} = (1 - B)y_t$$

$$y'_t = (1 - B)^d y_t$$

- **移动平均 MA(q)**：移动平均模型对过去的预测误差使用类似回归的模型。这里， ϵ 是时间 t 的白噪声， c 是常数， θ_s 是参数。

$$\hat{y}_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

结合上述所有三种类型的模型，得到了 ARIMA(p,d,q) 模型：

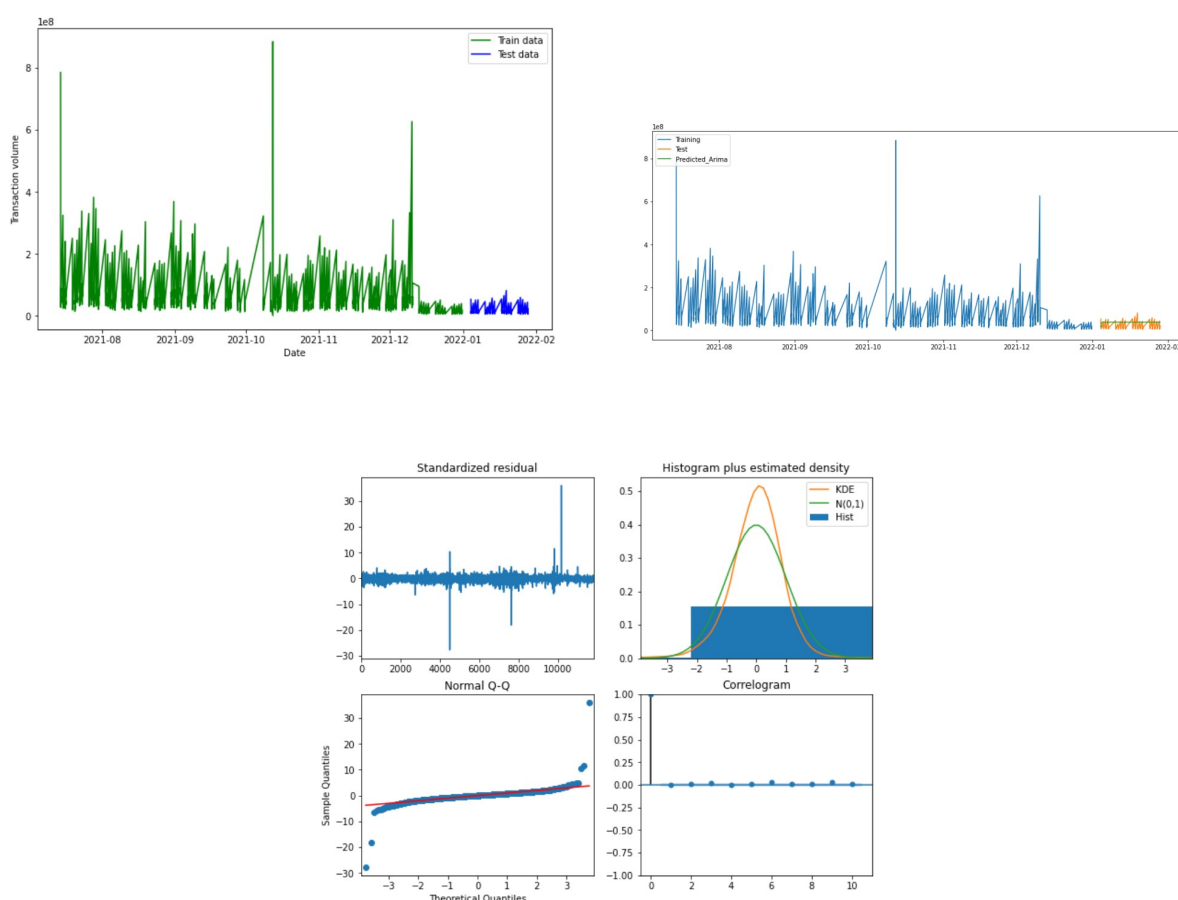
$$\hat{y}'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \dots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t$$

我们使用 Python 构建 ARIMA 模型并从中选择出最为合适的模型参数，最终模型选择的最优模型为 ARIMA(2,1,1)(0,0,0)[0]，将 2021 年 7 月 14 日至 2021 年 12 月 31 日的每 5 分钟“数字经济”板块指数为训练集，以 2022 年 1 月 4 日至 2022 年 1 月 28 日的每 5 分钟“数字经济”板块指数为测试集对模型进行训练。

(1) ARIMA 成交量预测模型结果

变量	数值
R^2	-958.9798%
平均绝对误差 MAE	24405767.1873
均方误差 MSE	632920326730052.8750
均方根误差 RMSE	25157907.8369
最大误差	43491878.2556

表 9 预测误差分析



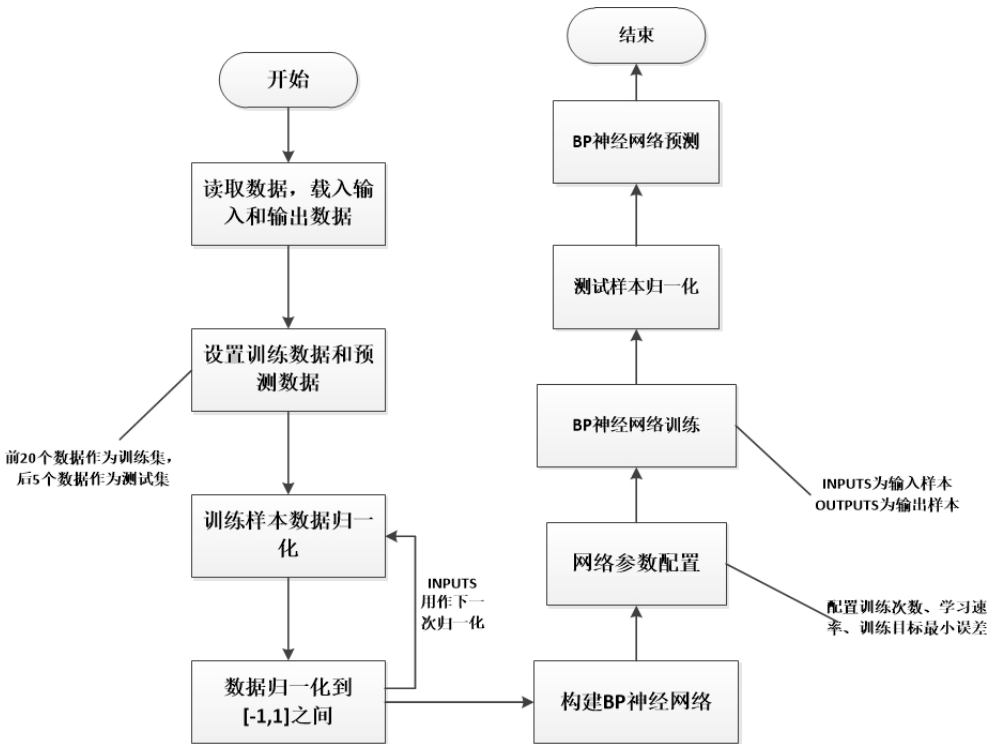
(2) ARIMA 成交量预测模型分析

在以上训练生成后的 ARIMA 模型中,观察结果图形可得模型中平均绝对误差 MAE 与均方误差 MSE 都较大,输入结果和期望输出结果二者相对误差均大于 1%,预测效果不理想。分析原因,可能是由于数据的分布并不具有很强的时间序列性,故并不采用 ARIMA 模型对“数字经济”板块指数每 5 分钟成交量进行预测。

6.2.2 基于 BP 神经网络的每 5 分钟成交量预测模型

BP 神经网络是基于的一种基于 Back-propagation 算法的多层前馈网络，使用梯度下降对人工神经网络进行监督学习，不需要特别提及要学习的功能的特征便能较好的映射输入与输出数据之间的关系。由于附件中所给数据量较多且数据呈现不明显时间序列关系，故我们优先使用 BP 神经网络来对“数字经济”板块指数每 5 分钟成交量预测。一般来说，BP 神经网络的算法步骤如下：

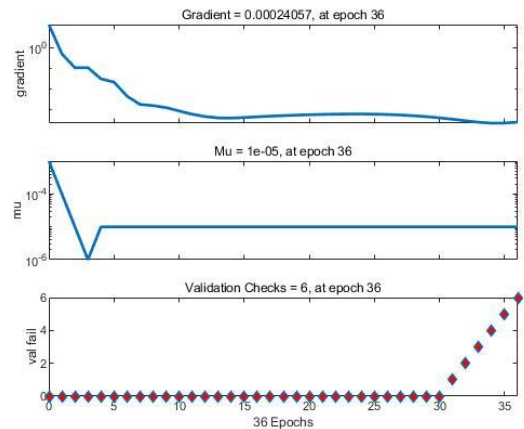
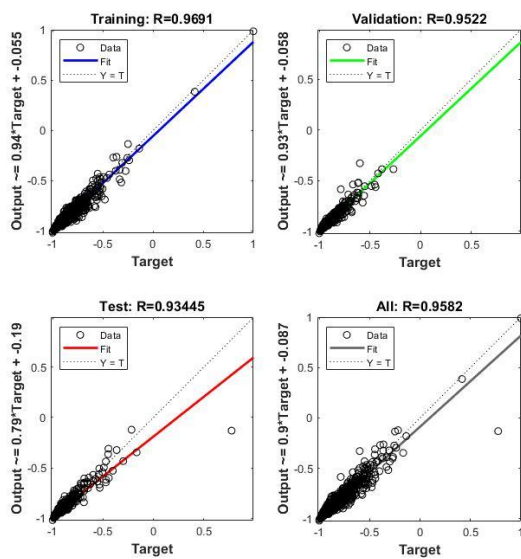
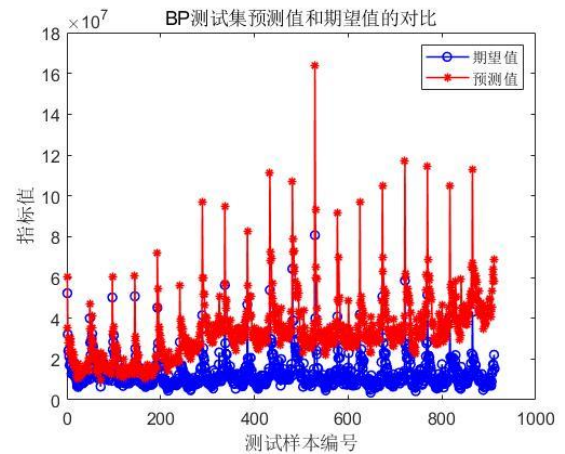
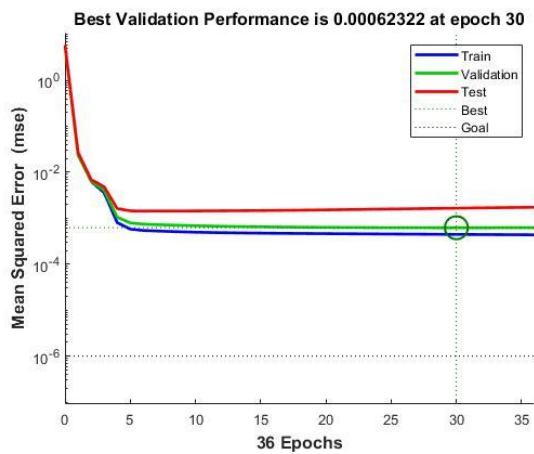
- **步骤 (1)：**初始化网络及学习参数，如设置网络初始权矩阵、学习因子等；
 - **步骤 (2)：**提供训练模式，训练网络直至满足所需的学习要求；
 - **步骤 (3)：**前向传播过程：对给定训练模式进行输入，计算网络的输出模式，并与期望模式比较，若有误差，则执行步骤(4), 否则返回步骤 (2)
 - **步骤 (4)：**反向传播过程：计算同一单元的误差，修正权值和阈值，返回步骤 (2)；
- 根据上述算法步骤，我们将附录 11.1.1 的数据进行拆分，以 2021 年 7 月 14 日至 2021 年 12 月 31 日的每 5 分钟“数字经济”板块指数为训练集，以 2022 年 1 月 4 日至 2022 年 1 月 28 日的每 5 分钟“数字经济”板块指数为测试集，对整个神经网络进行训练并检验。具体过程表述如下：



(1) BP 神经网络成交量预测模型结果

变量	数值
误差平方和 SSE	4.9043370767550510 ⁽¹⁷⁾
平均绝对误差 MAE	20459749.7772
均方误差 MSE	537756258416123.4
均方根误差 RMSE	23189572.1913
平均百分比误差 MAPE	181.3468%
相关系数 R	0.7609

表 10 预测误差分析



(2) BP 神经网络成交量预测模型分析

经程序运行后的模型相关系数为 0.7609，且从测试集预测值与期望值的对比中可以看出，模型能够从众多的数据中学习到关键特征。模型预测的具体结果详细见附录 11.2.2。

七、问题三的模型建立与求解

7.1 数据的处理与预处理

通过问题一与问题二的求解，我们获得了一个包含所有与“数字经济”板块有关指标的数据集。在此问题中，我们利用 Python 中的 pandas 库对该数据集进行清洗，并以时间作为数据的索引项，对空白值进行切除与拼接，获得附录 11.1.2 中的数据。

7.2 基于 LSTM 的收盘价预测模型

长短期记忆网络——通常称为“LSTM”——是一种特殊的 RNN，能够学习长期依赖关系。LSTM 是一种用于深度学习的人工循环神经网络，可以处理整个数据序列。由于该模型能够学习长期观察序列，LSTM 已成为时间序列预测的趋势方法。

LSTM 模块（或单元）具有 5 个基本组件，可以对长期和短期数据进行建模。

- 细胞状态 (c_t)：这表示存储短期记忆和长期记忆的细胞的内部记忆

$$c_t = f_{-t}c_{-t-1} + i_t\tilde{c}_t$$

$$\tilde{c}_t = \sigma(W_{cx}x_t + W_{ch}h_{t-1} + b_c)$$

- 隐藏状态 (h_t)：这是根据当前输入、先前隐藏状态和当前单元格输入计算的输出状态信息，最终使用这些信息来对数据进行预测处理。此外，隐藏状态可以决定只检索存储在单元状态中的短期或长期或两种类型的记忆，以进行下一次预测。

$$h_t = o_t \tanh(c_t)$$

- 输入门 (i_t)：决定有多少信息从当前输入流到细胞状态

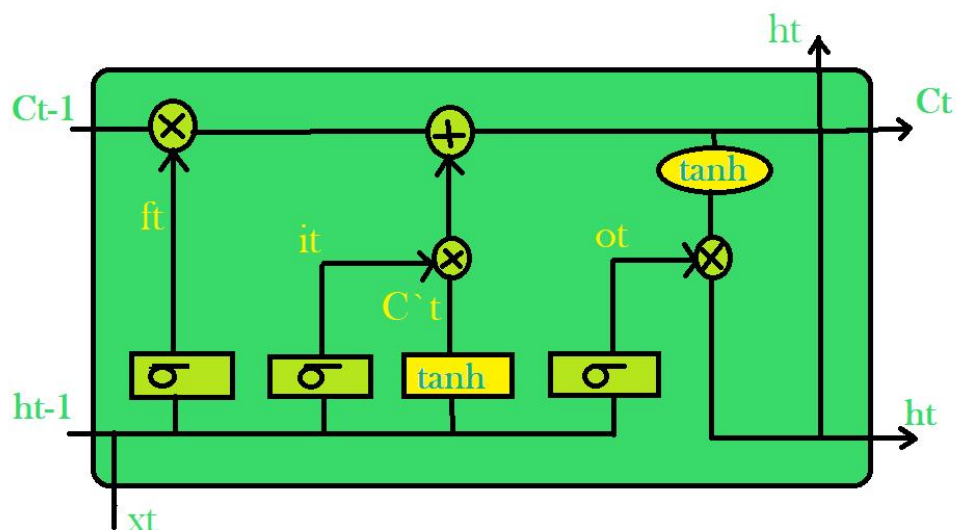
$$i_t = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + b_i)$$

- 遗忘门 (f_{-t})：决定有多少来自当前输入和前一个单元状态的信息流入当前单元状态

$$f_{-t} = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

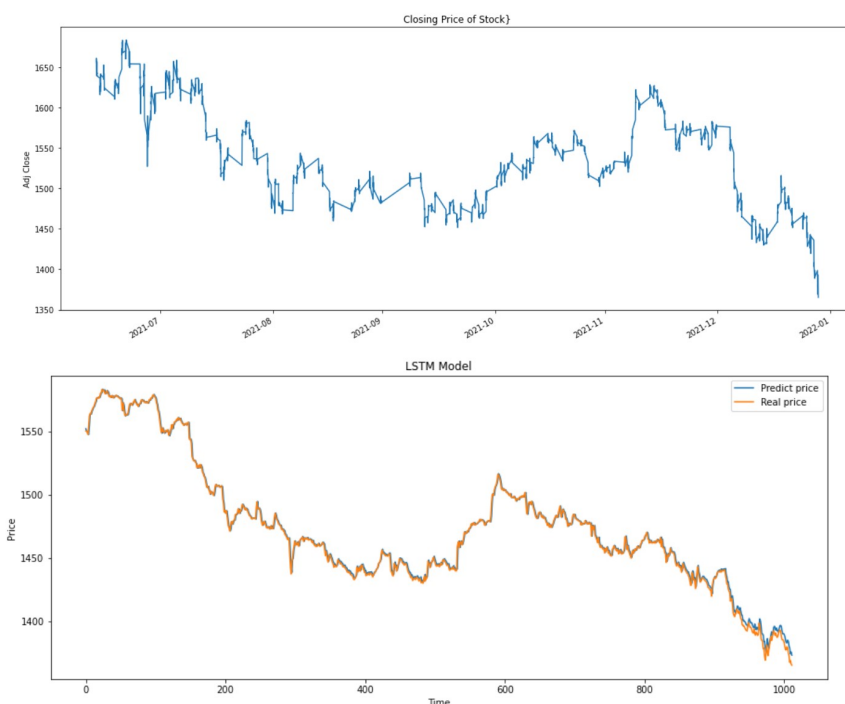
- 输出门 (o_{-t})：决定有多少信息从当前单元状态流入隐藏状态，这样如果需要 LSTM 只能选择长期记忆或短期记忆和长期记忆

$$o_{-t} = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$



我们使用附录 11.1.2 中的包含所有与“数字经济”板块有关指标的数据集，构建 LSTM 时间预测模型，以 2021 年 7 月 14 日至 2021 年 12 月 31 日的每 5 分钟“数字经济”板块指数构建训练集，以 2022 年 1 月 4 日至 2022 年 1 月 28 日的每 5 分钟“数字经济”板块指数构建测试集。设置 `batch_size=1`, `epochs=100`，对模型进行训练。

(1) LSTM 的收盘价预测模型结果



(2) GRU 的收盘价预测模型结果分析与比较

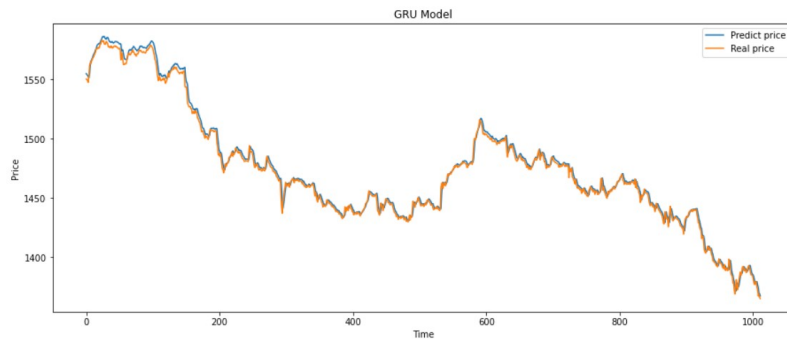
为了充分的了解评估 LSTM 模型的好坏，这里我们再次引入与 LSTM 模型非常相似的 GRU 模型进行训练。门控循环单元 (GRU) 是循环神经网络中的门控机制，由 Kyunghyun Cho 等人于 2014 年引入。GRU 就像一个带有遗忘门的长短期记忆 LSTM，

但参数比 LSTM 少，因为它没有输出门。

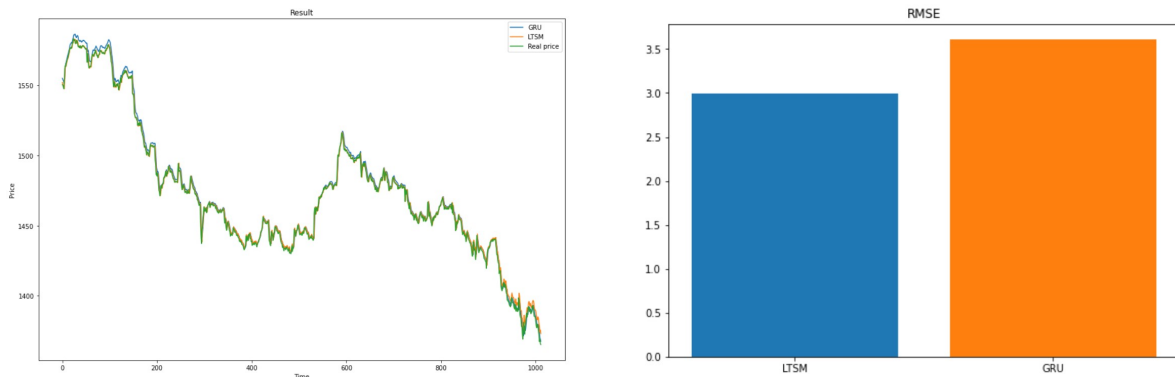
$$\begin{aligned} z^u &= \sigma(w_u * [h^{t-1}, x^t] + b_u) \\ z^r &= \sigma(w_r * [h^{t-1}, x^t] + b_r) \\ z &= \sigma(w * [h^{t-1} \odot z^r, x^t] + b) \\ h^t &= z^u \odot c^{t-1} + (1 - z^u) \odot z \\ y^t &= \sigma(w' h^t) \end{aligned}$$

其中 z^u 表示更新门， z^r 表示重置门， h^t 输出 1， y^t 输出 3。

GRU 的收盘价预测模型结果：



将两者模型训练后的结果进行比较得到如下结果：



从图中可以看出，LSTM 与 GRU 对模型的拟合效果均较好，其中 LSTM 的 RSE 为 2.991437，GRU 的 RSE 为 3.607073。LSTM 的拟合效果优于 GRU，在此，我们选择 **LSTM 模型** 对每 5 分钟的“数字经济”板块指数（收盘价）进行预测。

八、问题四的模型建立与求解

8.1 数据的预处理

我们利用在问题三中所建立的 LSTM 模型预测结果重新整合到数字经济”板块指数数据集中，并同时数据集进行清洗，得到同时包含有关预测结果和所有与“数字经济”板块有关指标的新数据集。（详情见附录 11.2.2）

8.2 交易策略的制定

在以“数字经济”板块指数为交易对象，每 5 分钟频率价格进行买卖交易时，存在着不同的交易策略。交易策略的不同会直接或间接地导致交易最后的总收益率、信息比率、最大回撤率。在本文中，我们主要提出以下几种交易策略：

策略一：仅考虑收盘价和交易量的 SMA

SMA 是一种简单地将某一股票在某一特定时间段的收盘价之和进行算术平均的方法，其中随着时间的推移将所求的平均值连成一条线便称之为 SMA。在计算 SMA 之后，我们建立一个策略，根据该策略生成买入或卖出信号。该策略为：当资产的收盘价低于其 SMA，并且交易量高于其 SMA 时，这是一个买入信号，反之亦然。接下来，将回测计算收盘价和成交量的 SMA 的天数，最终采用具有最佳夏普比率的值。

策略二：考虑短期和长期 SMA

第二种策略是创建一个短期 SMA（例如：10-d SMA）和一个较长的（60-d SMA），然后分析两者之间的交叉以生成买入/卖出信号：如果短期均线低于长期均线，则表示趋势正在下跌，因此是卖出信号，也称为死亡交叉。否则，趋势向上移动，这是一个买入信号，也称为黄金交叉。

策略三：布林带策略 (BB)

布林带策略 (BB) 是一种广泛使用的交易指标，简单但利润丰厚。它基本上是根据超买/超卖区域生成买入/卖出信号。布林带策略 (BB) 由三个主要部分组成：

- 中布林带 (MBB)，它是收盘价行为的 n 周期 SMA（通常我们取 $n=20$)：

$$MBB(PA, n) = SMA(PA, n)$$

- 上布林带 (UBB) 是一条比 MBB 高 m 个标准差的线（通常我们取 $m=2$)：

$$UBB(PA, n, m) = MBB(PA, n) + m * \sigma(PA, n)$$

- 下布林带 (LBB) 是一条低于 MBB 的 m 个标准差的线：

$$LBB(PA, n, m) = MBB(PA, n) - m * \sigma(PA, n)$$

当收盘价低于 LBB 时，我们做多头寸，如果收盘价超过 UBB，我们做空头寸。

策略四：随机震荡指标测量

随机震荡指标是一种动量交易策略，通过检测趋势价格反转来产生超买/超卖信号。它的工作原理是计算价格行为相对于交易日内最大低-高变化的变化。该策略包括两个主要信号：

%K 也称为快速振荡器，它计算实际收盘价相对于 14 个交易日内记录的最大变化的百分比变化：

$$\%K = \frac{C - L_d}{H_d - L_d} * 100$$

%D 或称为慢速振荡器，它是%K 的 3-d SMA，提供更平滑的信号表示：

$$\%D = SMA(\%K, 3)$$

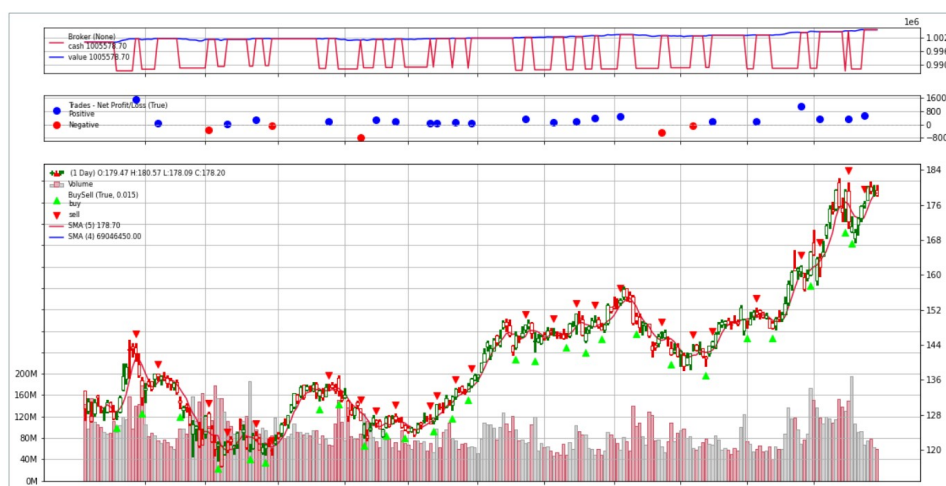
其中：C 为收盘价，Ld 和 Hd 分别为 d 个交易日内的最低价和最高价。

%K 介于 0 和 100% 之间时，我们认为如果它超过 80% 是超买区域，因此是卖出信号，当它低于 20% 或%K 超过%D 时，它是超卖区域，因此是买入信号。

8.3 交易模型的建立与求解

通过获取对数收益率计算出 Hurst 值，根据 Hurst 值的大小从上述四种决策中来选择不同的策略。

- Hurst < 0.5：随机震荡指标测量策略
- Hurst > 0.5：布林带策略 (BB)



其中，总收益率、信息比率、最大回撤率分别为 8.24%、45%、9.9%。

九、模型分析

9.1 模型优点

- 第一问中的典型相关性模型中我们将附件中所提供的各项指标均与“数字经济”板块进行了相关性分析，使得相关分析模型具有一定的严谨性和代表性。
- 第二、三问中的成交量预测模型中，我们均采用了不同的预测方法对数据进行拟合和预测。在第二问中，我们同时采用 ARIMA 时间序列预测模型与 BP 神经网络预测模型对“数字经济”板块指数每 5 分钟成交量进行分析。在第三问中，利用 LSTM 和 GRU 模型同时对每 5 分钟的“数字经济”板块指数（收盘价）进行预测。并在二三问中，都对模型的拟合度和结果进行比较。该方法在一定程度上弥补了由于数据并不全面的不足，同时也考虑到了单个模型对最终结果求解的局限性。

- 第四问中的交易模型中，我们根据参考文献和股市实际情况，提出了四种不同的交易策略，并根据不同的情况来对交易进行决策，具有一定的创新性与实际代表性。

9.2 模型缺点

- 我们并没有考虑宏观市场指标与汇率中的指标，忽略这些指标可能会导致我们的预测模型与实际结果之间存在偏差。
- 由于数据的局限性与缺失及时间有限，我们只是简单的用均值来对缺失的数据进行插入，并所提出的模型训练时长可能存在不足。

9.3 模型的改进

限于问题所给数据集规模总体来说不是很大，因此本文中的预测与评价模型可能会与实际间存在一些误差，未来可以通过更多的数据集（更多时间、更多的指标类）来对模型进行训练，进而才能够进一步提升模型预测的拟合优度和准确率。并通过增加训练时长及微调，来对模型进行进一步改进。

十、参考文献

- [1] 王鹏飞, 程华. 基于知识图谱的国内数字经济演化路径、热点及展望 [J/OL]. 科学与管理:1-12[2022-05-02].
- [2] 陈晓东, 刘冰冰. 数字经济推动区域协调发展的实证研究——基于 2013 ~ 2019 年中国省级面板数据 [J]. 经济研究参考,2022(04):78-97.
- [3] 王娟, 张蕴洁, 宋洁, 张平文. 中美欧数字经济与贸易的比较研究 [J/OL]. 西安交通大学学报 (社会科学版):1-15[2022-05-02].
- [4] 刘顿. 数字经济时代美好生活消费的功能价值及实践向度 [J/OL]. 社会科学家,2022(02):79-87[2022-05-02].
- [5] 邓峰, 冯福博, 杨小东. 市场分割、数字经济与区域创新效率 [J/OL]. 统计与决策,2022(09):17-20[2022-05-02].
- [6] 张茂军, 饶华城, 南江霞, 王国栋. 基于决策树的量化交易择时策略研究 [J/OL]. 系统工程:1-17[2022-05-02].
- [7] 潘晓展. 基于量化风格的再优化投资策略 [D]. 商务部国际贸易经济合作研究院,2021.
- [8] 王权鑫. 基于最大回撤率衡量风险的投资组合策略研究 [D]. 山东科技大学,2019.
- [9] 单超. 基于信息比率加权的因子选股策略绩效评价 [D]. 山东大学,2020.

十一、附录

11.1 数据集

11.1.1 问题一重要指标选取过程的数据集

一. 技术指标

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.976	19.993	0.015	8.932	84.000	664.072	0.000
2	0.629	0.653	0.310	2.447	65.000	566.316	0.000
3	0.511	0.353	0.513	1.828	48.000	464.291	0.001
4	0.460	0.268	0.694	1.426	33.000	357.192	0.065
5	0.306	0.103	0.881	0.801	20.000	244.000	0.711
6	0.169	0.029	0.972	0.400	9.000	123.000	0.933

表 11 典型相关性结果一

变量	1	2	3	4	5	6
VMA	-0.645	-1.274	-1.057	-1.441	2.967	0.551
VMACD	0.302	1.750	0.430	1.418	-2.512	-0.336
ARBR	0.096	0.836	1.208	0.160	-0.192	-0.120
OBV	-0.083	0.206	0.452	-1.231	1.697	1.748
BBI	-3.003	-0.881	-5.993	-4.949	-16.524	-2.841
DMA	-0.152	-0.008	3.107	6.781	9.296	-1.466
MA	0.264	-1.854	2.368	-5.315	7.563	-4.517
EXPMA	-0.600	10.004	9.683	19.778	6.676	6.163
MTM	0.067	0.151	-0.825	1.155	0.472	-0.219
MACD	1.859	-4.564	-8.626	-11.754	-10.306	1.647
BIAS	0.107	-1.735	-0.320	0.205	1.612	-0.764
KDJ	0.079	-0.186	0.820	0.552	.766	0.070
RSI	-0.146	-0.257	0.426	-2.005	-2.153	0.201
BOLL	2.269	-5.253	-2.989	-5.837	.142	-0.492

表 12 标准化典型相关系数结果一

二. 国际市场

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.965	13.712	0.034	15.955	66.000	1187.993	0.000
2	0.548	0.430	0.493	3.408	50.000	1015.840	0.000
3	0.422	0.217	0.705	2.276	36.000	837.422	0.000
4	0.304	0.102	0.858	1.472	24.000	650.269	0.069
5	0.222	0.052	0.945	0.922	14.000	450.000	0.535
6	0.079	0.006	0.994	0.235	6.000	226.000	0.965

表 13 典型相关性结果二

变量	1	2	3	4	5	6
道琼斯工业平均指数	-0.952	0.953	-0.944	0.135	-2.934	-0.888
纳斯达克综合指数	0.185	-0.708	0.406	2.301	-3.390	-0.958
标准普尔 500 指数	0.955	-0.958	0.806	-2.805	9.452	-0.274
美国证交所	0.620	1.378	0.657	-0.546	-2.447	-1.109
恒生指数	-0.187	-1.258	0.254	-0.578	0.164	-1.237
东京日经 225 指数	-0.157	-0.103	-0.336	-0.643	0.389	0.820
伦敦金融时报 100 指数	-0.040	-1.534	-0.969	0.708	-1.613	1.691
法国巴黎 CAC40 指数	-0.778	2.291	-2.044	-1.340	1.442	-0.139
荷兰 AEX 指数	0.245	-0.102	2.183	-3.325	-3.318	-0.029
俄罗斯 RTS 指数	-0.305	0.414	-0.572	0.871	0.968	0.795
意大利 MIB 指数	0.709	-2.830	0.235	3.843	1.491	0.037

表 14 标准化典型相关系数结果二

三. 国内市场

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.979	23.084	0.022	19.012	72.000	1338.749	0.000
2	0.536	0.403	0.522	3.133	55.000	1142.265	0.000
3	0.386	0.175	0.732	2.009	40.000	938.451	0.000
4	0.323	0.117	0.860	1.418	27.000	724.930	0.079
5	0.166	0.028	0.961	0.630	16.000	498.000	0.860
6	0.110	0.012	0.988	0.441	7.000	250.000	0.876

表 15 典型相关性结果三

变量	1	2	3	4	5	6
成交量上证综合指数	-0.014	0.483	1.947	2.151	-0.009	0.172
成交金额上证综合指数	0.073	0.009	-2.315	-1.687	0.399	-0.582
沪市股票流通市值	-0.616	-9.765	-4.947	4.969	1.457	0.760
深市股票流通市值	2.654	7.478	5.880	-8.805	-4.266	-0.087
沪深 300 指数	-0.830	4.492	3.207	-14.403	-7.783	1.639
上证综合指数	13.440	439.938	-171.243	113.539	230.306	-44.997
中证 500 指数	-0.550	-3.867	-1.393	2.238	-2.121	-5.335
创业板指数	0.294	2.153	0.578	-0.710	-0.844	-3.711
上证 50 指数	0.430	-2.917	-5.508	11.441	0.864	-7.115
上证 A 股指数	-12.782	-433.693	174.735	-117.077	-226.879	46.935
深证成份指数	0.309	-6.833	0.940	7.616	6.693	3.308
深证综合指数	-1.761	1.359	-4.370	2.826	-3.146	1.484

表 16 标准化典型相关系数结果三

四. 其他指标

	相关性	特征值	威尔克统计	F	分子自由度	分母自由度	显著性
1	0.962	12.277	0.062	31.615	30.000	942.000	0.000
2	0.354	0.143	0.819	2.438	20.000	783.673	0.000
3	0.184	0.035	0.936	1.325	12.000	627.335	0.199
4	0.176	0.032	0.969	1.274	6.000	476.000	0.268
5	0.020	0.000	1.000				

表 17 典型相关性结果四

变量	1	2	3	4	5
数字媒体	-0.589	1.194	0.466	0.630	1.742
数字孪生	-0.085	-0.022	1.368	0.899	2.301
快手概念	0.215	-0.210	-0.135	0.626	-1.730
互联网电商	-0.482	-1.255	0.326	0.529	-0.034
互联网	0.426	-0.386	-1.416	-0.465	-0.272

表 18 标准化典型相关系数结果四

11.1.2 问题二与“数字经济”板块有关指标的数据集

时间	开盘价	收盘价	最高价	最低价	成交量	成交额	数字媒体	上证综合指数	上证 A 股指数	互联网电商	道琼斯工业平均指数	标准普尔 500 指数	BOLL	BBI
2021/07/14 09:35:00	1650.743	1656.128	1656.128	1645.318	7.85E+08	1.07E+10	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 09:40:00	1656.456	1660.884	1660.884	1655.757	2.95E+08	5.12E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 09:45:00	1661.482	1651.137	1661.547	1651.137	1.7E+08	3.55E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 09:50:00	1650.107	1651.691	1652.069	1647.201	2.02E+08	4.25E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 09:55:00	1651.927	1650.045	1654.952	1649.144	1.25E+08	3.11E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:00:00	1649.982	1646.225	1650.189	1646.225	1.52E+08	2.71E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:05:00	1646.111	1646.062	1647.367	1644.521	1.19E+08	2.41E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:10:00	1645.903	1640.648	1646.123	1640.648	99206800	2.29E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:15:00	1640.374	1640.671	1643.837	1640.138	81490800	2.02E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:20:00	1640.454	1643.678	1644.983	1640.252	69059600	1.7E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:25:00	1643.779	1645.304	1646.165	1643.678	79064400	1.7E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:30:00	1645.355	1641.02	1645.814	1640.593	56399400	1.4E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:35:00	1640.985	1640.008	1643.313	1639.733	51665200	1.26E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:40:00	1640.57	1639.552	1641.574	1639.548	46005300	1.14E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:45:00	1639.525	1643.733	1643.735	1639.074	52597700	1.24E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:50:00	1643.424	1643.074	1643.424	1638.869	66550200	1.51E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 10:55:00	1643.601	1644.455	1645.535	1643.601	47735600	1.12E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:00:00	1644.269	1645.81	1645.81	1643.83	37730700	9.49E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:05:00	1645.638	1644.555	1646.776	1644.313	38918100	9.09E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:10:00	1644.606	1644.905	1644.905	1643.474	29796700	7.97E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:15:00	1644.499	1642.182	1645.235	1641.638	35586400	8.66E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:20:00	1642.16	1644.631	1644.631	1641.632	38251300	8.46E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:25:00	1644.746	1643.677	1645.158	1643.435	33545800	7.37E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 11:30:00	1644.05	1643.554	1644.347	1642.629	26290100	6.24E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:05:00	1643.641	1642.176	1644.534	1642.154	60836100	1.34E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:10:00	1641.991	1641.139	1641.991	1639.963	57738200	1.06E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:15:00	1641.096	1645.067	1645.397	1641.096	40563100	9.21E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:20:00	1644.663	1645.788	1645.788	1644.205	42514400	9.89E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:25:00	1646.071	1647.63	1648.55	1645.659	61643800	1.4E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:30:00	1647.825	1646.778	1648.655	1646.55	50300700	1.2E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:35:00	1646.811	1652.576	1652.576	1646.769	50909600	1.12E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:40:00	1652.233	1655.323	1655.39	1652.233	48926700	1.33E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:45:00	1655.325	1656.271	1656.271	1653.769	39772500	1.24E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:50:00	1656.06	1656.569	1657.275	1654.37	40829200	1.18E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 13:55:00	1656.583	1654.998	1656.886	1654.195	37308100	1.12E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:00:00	1655.141	1652.053	1655.708	1652.023	36096800	1.05E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:05:00	1652.078	1646.572	1652.294	1646.572	39504700	1.02E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:10:00	1646.544	1645.906	1646.544	1645.383	49805000	1.11E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:15:00	1646.078	1644.78	1646.091	1644.096	43992800	1.03E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:20:00	1644.851	1641.592	1644.862	1640.262	51171600	1.34E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:25:00	1642.037	1644.381	1644.542	1642.037	40523800	9.85E+08	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:30:00	1644.493	1642.717	1645.045	1642.58	45824100	1.13E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:35:00	1642.492	1644.085	1644.887	1642.263	47517300	1.15E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:40:00	1644.138	1642.834	1644.422	1642.834	55308300	1.12E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:45:00	1642.839	1640.564	1642.839	1640.564	63891000	1.38E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:50:00	1640.483	1640.382	1641.387	1639.54	66384300	1.47E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 14:55:00	1640.353	1638.781	1640.568	1638.692	88430000	1.93E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348
2021-07-14 15:00:00	1638.71	1639.159	1639.159	1638.426	71267700	1.41E+09	2470.63	3528.501	3698.173	22369.93	35084.53	4374.3	1620.318	1643.348

11.2 结果集

11.2.1 问题一与“数字经济板块”有关的八个重要指标

序号	指标名
1	BBI
2	BOLL
3	道琼斯工业平均指数
4	标准普尔 500 指数
5	上证综合指数
6	上证 A 股指数
7	数字媒体
8	互联网电商

11.2.2 问题二 “数字经济板块” 与预测结果集

时间	开盘价	收盘价	最高价	最低价	成交量	成交额	数字媒体	上证综合指数	上证 A 股指数	互联网电商	道琼斯工业平均指数	标准普尔 500 指数	BOLL	BBI	预测结果
44565.4	1584.952	1575.839	1586.753	1575.086	52261199	2.98E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	52261199
44565.4	1576.138	1572.615	1578.148	1572.1	32032367	1.79E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	32032367
44565.41	1572.85	1570.878	1574.837	1570.869	24013795	1.41E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	24013795
44565.41	1570.824	1566.618	1571.492	1566.618	24464949	1.49E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	24464949
44565.41	1566.403	1565.711	1567.082	1564.423	21194010	1.17E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	21194010
44565.42	1565.726	1562.212	1565.726	1561.649	16934790	1.08E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	16934790
44565.42	1562.53	1557.361	1562.79	1557.361	17331600	1.26E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	17331600
44565.42	1557.213	1553.419	1557.213	1552.801	19476130	1.37E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	19476130
44565.43	1553.349	1548.907	1553.457	1547.972	17510690	1.28E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	17510690
44565.43	1549.074	1549.357	1551.959	1549.054	12870860	9.41E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	12870860
44565.43	1549.398	1552.606	1552.896	1548.171	12939260	9.07E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	12939260
44565.44	1552.944	1550.43	1554.99	1550.2	19212680	1.09E+09	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	19212680
44565.44	1550.393	1548.786	1551.111	1548.586	13070790	9.01E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	13070790
44565.44	1548.999	1549.125	1551.224	1548.453	16000880	9.85E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	16000880
44565.45	1548.871	1549.726	1550.951	1548.717	12114410	8.51E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	12114410
44565.45	1549.817	1550.747	1550.941	1549.693	11926030	7.49E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	11926030
44565.45	1550.626	1549.603	1551.324	1549.277	11315530	6.54E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	11315530
44565.46	1549.548	1551.089	1551.795	1549.34	12868670	7.07E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	12868670
44565.46	1551.476	1548.975	1551.476	1548.846	10782520	7.1E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	10782520
44565.47	1548.831	1546.617	1548.831	1546.337	10829420	6.95E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	10829420
44565.47	1546.75	1547.944	1548.631	1546.565	9052740	5.99E+08	2732.5	3632.329	3806.498	17836.12	36407.11	4793.54	1565.28	1557.967	9052740

11.2.3 问题三 “数字经济板块” 与预测结果集

时间	开盘价	收盘价	最高价	最低价	成交量	成交额	数字媒体	上证综合指数	上证 A 股指数	互联网电商	道琼斯工业平均指数	标准普尔 500 指数	BOLL	BBI	预测结果
2021-12-29 14: 45: 00	1551.796	1550.253	1552.562	1550.253	9844810	6.02E+08	2407.6	3597	3769.551	17144.59	36338.3	4793.06	1561.579	1566.318	1551.891
2021-12-29 14: 50: 00	1550.216	1549.869	1550.361	1549.771	10570580	6.32E+08	2407.6	3597	3769.551	17144.59	36338.3	4793.06	1561.579	1566.318	1550.187
2021-12-29 14: 55: 00	1549.711	1548.792	1549.912	1548.624	12674190	7.64E+08	2407.6	3597	3769.551	17144.59	36338.3	4793.06	1561.579	1566.318	1549.777
2021-12-29 15: 00: 00	1548.474	1547.429	1548.551	1547.429	9600540	4.92E+08	2407.6	3597	3769.551	17144.59	36338.3	4793.06	1561.579	1566.318	1548.702
2021-12-30 9: 35: 00	1545.864	1553.729	1556.732	1545.852	36654402	2.21E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1547.299
2021-12-30 9: 40: 00	1553.155	1561.702	1562.349	1553.155	26240443	1.45E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1553.869
2021-12-30 9: 45: 00	1561.426	1563.777	1563.986	1559.807	24296883	1.25E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1562.183
2021-12-30 9: 50: 00	1563.616	1563.358	1566	1562.599	23606763	1.31E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1564.219
2021-12-30 9: 55: 00	1563.56	1565.408	1566.223	1563.56	20735584	1.26E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1563.555
2021-12-30 10: 00: 00	1565.229	1566.777	1567.542	1564.3	17948565	1.11E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1565.614
2021-12-30 10: 05: 00	1566.607	1568.163	1570.167	1566.26	18186530	1.23E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1567.096
2021-12-30 10: 10: 00	1568.513	1568.828	1570.964	1567.959	19154090	1.27E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1568.501
2021-12-30 10: 15: 00	1568.774	1570.092	1570.707	1568.676	14768040	9.19E+08	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1569.155
2021-12-30 10: 20: 00	1570.368	1572.051	1572.546	1570.093	16086340	1.06E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1570.414
2021-12-30 10: 25: 00	1572.026	1572.944	1573.652	1571.647	18488440	1E+09	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1572.411
2021-12-30 10: 30: 00	1572.914	1575.885	1575.885	1572.855	14809230	9.82E+08	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1573.302
2021-12-30 10: 35: 00	1575.784	1575.843	1576.946	1575.466	13389410	9.69E+08	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1576.258
2021-12-30 10: 40: 00	1575.44	1575.908	1577.318	1575.254	10328440	9.22E+08	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1576.212
2021-12-30 10: 45: 00	1575.984	1576.846	1576.926	1575.984	8602510	6.26E+08	2569.5	3619.189	3792.798	17227.74	36585.06	4778.73	1562.426	1567.61	1576.18

11.3 模型代码

11.3.1 问题二 ARIMA 预测模型 Python 代码

```

pip install pmdarima
pip install termcolor
import os
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from termcolor import colored
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from pmdarima.arima import auto_arima
from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
from sklearn.metrics import max_error
from sklearn.metrics import r2_score
df = pd.read_excel('/content/数字板块与指数
(2).xlsx',sheet_name=0,index_col='时间')
df.sort_index(inplace=True)
df.dropna(inplace=True)
df = df.drop(columns=['指数代码'])

```

```

#拆分数数据集为测试集和数据集
train_data, test_data = df[:-912], df[-912:]
plt.figure(figsize=(10,6))
plt.xlabel('Date')
plt.ylabel('Transaction volume')
plt.plot(train_data['成交量'], 'green', label='Train data')
plt.plot(test_data['成交量'], 'blue', label='Test data')
plt.legend()
train_data_endog = train_data['成交量'].values
test_data_endog = test_data['成交量'].values
from pmdarima.arima.utils import ndiffs
y = train_data_endog
print('adf:', ndiffs(y, test='adf'))
print('kpss:', ndiffs(y, test='kpss'))
print('pp:', ndiffs(y, test='pp'))
from statsmodels.graphics.tsaplots import plot_pacf
columns = ['成交量']
for i in columns:
    plot_pacf(df[i].diff().dropna(),lags=15,title=i)
    plt.show()
    print('')
from statsmodels.graphics.tsaplots import plot_acf
columns = ['成交量']
for i in columns:
    plot_acf(df[i].diff().dropna(),lags=15,title=i)
    plt.show()
    print('')
train_data_endog = train_data_endog.reshape(-1, 1)
test_data_endog = test_data_endog.reshape(-1, 1)
from sklearn.preprocessing import PowerTransformer
pt_endog = PowerTransformer(method='yeo-johnson')
train_data_endog = pt_endog.fit_transform(train_data_endog)
test_data_endog = pt_endog.transform(test_data_endog)
model_autoARIMA = auto_arima(train_data_endog , start_p=0, start_q=0,
                             test='adf',
                             max_p=2, max_q=2,

```

```

        m=1,
        d=None,
        seasonal=False,
        trace=True,
        error_action='ignore',
        suppress_warnings=True,
        stepwise=True)
model_autoARIMA.plot_diagnostics(figsize=(10,8))
plt.show()
prediction_ARIMA = pd.DataFrame(model_autoARIMA.predict(n_periods = 912),
index=test_data.index)
prediction_ARIMA.columns = ['Predictions_ARIMA']
prediction_rescaled_ARIMA =
    pt_endog.inverse_transform(prediction_ARIMA.values)
prediction_ARIMA['Predictions_scaled_ARIMA'] = prediction_rescaled_ARIMA
prediction_ARIMA
plt.figure(figsize=(15,6))
plt.plot(train_data['成交量'],label="Training")
plt.plot(test_data['成交量'],label="Test")
plt.plot(prediction_ARIMA.Predictions_scaled_ARIMA,
    label="Predicted_Arima")
plt.legend(loc = 'upper left')
plt.show()

```

11.3.2 问题二 BP 神经网络预测 MATLAB 代码

```

def evaluate_prediction(predicted, actual, model_name):
    mse = mean_squared_error(predicted, actual)
    rsme = np.sqrt((mean_squared_error(predicted, actual)))
    mae = mean_absolute_error(actual, predicted)
    r2 = r2_score(actual, predicted)
    max_err = max_error(actual, predicted)
    print(colored(model_name + ' performance:', 'red'))
    print('R^2: {:.4f} %'.format(r2 * 100))
    print('Mean Absolute Error: {:.4f}'.format(mae))
    print('Mean Squared Error: {:.4f}'.format(mse))
    print('Root Mean Squared Error: {:.4f}'.format(rsme))

```

```

    print('Max_error: {:.4f}'.format(max_err))
    print('')
    return
evaluate_prediction(prediction_ARIMA.Predictions_scaled_ARIMA,
    test_data['成交量'], 'ARIMA')
clear
close
clc
formatshort
mydata=xlsread('数据.xlsx','Sheet7','A1:N6240');
input= mydata(:,1:trainN-1);
output = mydata(:,trainN);
N =length(output);
testN = 20;
trainN = N - testN;
input_train =input(1:trainN,:);
output_train = output(1:trainN);
input_test =input(trainN+1:N,:);
output_test = out(train+1:N);
[inputn,inputps] = mapminmax(input_train,0,1);
[outputn,outputps] = mapminmax(out_train,0,1);
inputn_test = mapminmax('apply',input_test,inputps);
inputnum =size(input,2);
outputnum =size(output,2);
MSE = 1e + 5
transform_func = {'tansig','purelin'};
train_func ='trainlm';
forhiddennum =
fix(sqrt(inputnum+outputnum))+1:fix(sqrt(inputnum+outputnum))+10
net = newff(inputn,outputn,hiddennum,transform_func,train_func);
net.trainParam.epochs = 100000;
net.trainParam.lr = 0.01;
net.trainParam.goal = 0.0000000001;
net = train(net,inputn,outputn);
an0 = sim(net,inputn);

```

```

mse0 = mse(outputn,an0);
ifmse0 < MSE
MSE = mse0;
hiddennum_best = hiddennum;
end
end
net=newff(inputn,outputn,hiddennum_best,transform_func,train_func);
net.trainParam.epochs=1000;
net.trainParam.lr=0.01;
net.trainParam.goal=0.000001;
net=train(net,inputn,outputn);
n=sim(net,inputn_test);
test_simu=mapminmax('reverse',an,outputps);
error=test_simu-output_test;
figure
plot(output_test,'bo-','linewidth',1.2)
holdon
plot(test_simu,'r*-','linewidth',1.2)
[~,len]=size(output_test);
SSE1=sum(error.^2);
MAE1=sum(abs(error))/len;
MSE1=error*error'/len;
RMSE1=MSE1^(1/2);
MAPE1=mean(abs(error./output_test));
r=corrcoef(output_test,test_simu);
R1=r(1,2);
fori=1:len
disp([i,output_test(i),test_simu(i),error(i)])
end

```

11.3.3 问题三 LSTM 时间序列预测模型 Python 代码

```

import seaborn as sns
import pandas as pd
import talos as ta
from numpy import zeros, newaxis
from matplotlib import pyplot as plt

```



```

from joblib import dump, load
from keras.utils.vis_utils import plot_model
from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler, PowerTransformer,
    StandardScaler
from sklearn.metrics import mean_squared_error, r2_score, max_error,
mean_absolute_error
from tensorflow.keras.optimizers import Adam, Nadam
from tensorflow.keras.utils import plot_model
from tensorflow.keras import Sequential, layers, callbacks
from tensorflow.keras.layers import Input, Dense, LSTM, Dropout, GRU,
Bidirectional, SimpleRNN, Conv1D, MaxPooling1D, Flatten
from keras.utils.vis_utils import plot_model
from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler, PowerTransformer,
    StandardScaler
from sklearn.metrics import mean_squared_error, r2_score, max_error,
mean_absolute_error
from tensorflow.keras.optimizers import Adam, Nadam
from tensorflow.keras.utils import plot_model
from tensorflow.keras import Sequential, layers, callbacks
from tensorflow.keras.layers import Input, Dense, LSTM, Dropout, GRU,
Bidirectional, SimpleRNN, Conv1D, MaxPooling1D, Flatten
df = pd.read_excel('数字板块与指数 (2).xlsx', index_col='时间')
df.sort_index(inplace=True)
df.dropna(inplace=True)
df = df.drop(columns=['指数代码'])
data = df
data.info
plt.figure(figsize=(15, 6))
plt.subplots_adjust(top=1.25, bottom=1.2)
data['收盘价'].plot()
plt.ylabel('Adj Close')
plt.xlabel(None)
plt.title("Closing Price of Stock}")

```

```

plt.tight_layout()
data = data.filter(['收盘价'])
dataset = data.values
training_data_len = int(np.ceil(5228))
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
train_data = scaled_data[0:int(training_data_len), :]
x_train = []
y_train = []
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
test_data = scaled_data[training_data_len - 60: , :]
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
tf.random.set_seed(1234)
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape=
    (x_train.shape[1],1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=1e-2,
    decay_steps=1427,

```

```

    decay_rate=0.9)
opt = tf.keras.optimizers.Adam(learning_rate = lr_schedule)
model.compile(optimizer=opt, loss='mean_squared_error')
model.fit(x_train, y_train, batch_size=1, epochs=30)
predictions_lstm = model.predict(x_test)
predictions_lstm = scaler.inverse_transform(predictions_lstm)
rmse_lstm = np.sqrt(np.mean(((predictions_lstm - y_test) ** 2)))
print("LSTM model rmse rate: {}".format(rmse_lstm))
plt.figure(figsize=(15, 6))
plt.plot(predictions_lstm, label="Predict price")
plt.plot(y_test, label="Real price")
plt.title("LSTM Model")
plt.xlabel("Time")
plt.ylabel("Price")
plt.legend()
plt.show()

```

11.3.4 问题三 GRU 时间序列预测模型 Python 代码

```

import tensorflow as tf
from tensorflow.keras import layers
tf.random.set_seed(1234)
model = tf.keras.Sequential([
    layers.GRU(128, return_sequences=True),
    layers.Dropout(0.2),
    layers.GRU(64),
    layers.Dropout(0.2),
    layers.Dense(1)
])
opt = tf.keras.optimizers.Adam(learning_rate = 0.0001)
model.compile(optimizer = opt, loss='mean_squared_error')
# 模型的训练
model.fit(x_train, y_train, batch_size=1, epochs=30)
predictions_gru = model.predict(x_test)
predictions_gru = scaler.inverse_transform(predictions_gru)
rmse_gru = np.sqrt(np.mean(((predictions_gru - y_test) ** 2)))
print("GRU mode rmse rate: {}".format(rmse_gru))

```

```
plt.plot(predictions_gru, label="Predict price")
plt.plot(y_test, label="Real price")
plt.title("GRU Model")
plt.xlabel("Time")
plt.ylabel("Price")
plt.legend()
plt.show()
```

11.3.5 问题三 LSTM 与 GRU 模型效果比较 Python 代码

```
plt.figure(figsize=(15, 10))
plt.plot(predictions_gru, label="GRU")
plt.plot(predictions_lstm, label="LSTM")
plt.plot(y_test, label="Real price")
plt.title("Result")
plt.xlabel("Time")
plt.ylabel("Price")
plt.legend()
plt.show()

import pandas as pd
data = {'Model': ['LSTM', 'GRU'], 'RMSE': [rmse_lstm, rmse_gru]}
df = pd.DataFrame(data)
print (df)

plt.bar("LSTM", rmse_lstm)
plt.bar("GRU", rmse_gru)
#plt.bar("MLP", rmse_mlp)
plt.title("RMSE")
plt.show()
```

11.3.5 问题四 Python 代码

```
!pip install yfinance pandas-ta backtrader
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime
import pandas_ta as ta
```

```

import backtrader as bt
import backtrader.analyzers as btanalyzers
import backtrader.feeds as btfeeds
import backtrader.strategies as btstrats
from IPython.display import display
class TradeLogger(bt.analyzers.Analyzer):
    def start(self):
        super(TradeLogger, self).start()
    def create_analysis(self):
        self.rets = []
        self.vals = dict()
    def notify_trade(self, trade):
        if trade.isclosed:
            self.vals = {'Date': self.strategy.datetime.datetime(),
                        'Gross PnL': round(trade.pnl, 2),
                        'Net PnL': round(trade.pnlcomm, 2),
                        'Trade commission': trade.commission,
                        'Trade duration (in days)': (trade.dtclose -
trade.dtopen)
                        }
            self.rets.append(self.vals)
    def get_analysis(self):
        return self.rets
def set_and_run(data, strategy, startcash, commission, stake):
    cerebro = bt.Cerebro()
    cerebro.addstrategy(strategy)
    cerebro.adddata(data)
    cerebro.broker.setcash(startcash)
    print("Starting Portfolio Value: {}".format(cerebro.broker.getvalue()))
    cerebro.broker.setcommission(commission)
    cerebro.addsizer(bt.sizers.FixedSize, stake=stake)
    cerebro.addanalyzer(bt.analyzers.SharpeRatio,
                        timeframe=bt.TimeFrame.Days,
compression=1, factor=365, annualize=True)
    cerebro.addanalyzer(TradeLogger, _name="trade_logger")
    results = cerebro.run()

```

```

print("Final Portfolio Value: {}".format(cerebro.broker.getvalue()))
print("Sharpe Ratio:
{}".format(results[0].analyzers.sharperatio.get_analysis()['sharperatio']))
display(pd.DataFrame(results[0].analyzers.trade_logger.get_analysis()))
plt.rcParams['figure.figsize'] = (16, 8)
fig = cerebro.plot(barupfill=False,
                    bardownfill=False,
                    style='candle',
                    plotdist=0.5,
                    volume=True,
                    barup='green',
                    valuetags=False,
                    subtxtsize=8)

class FirstStrategy(bt.Strategy):
    params = (('close_period', 5),
              ('volume_period', 4)
    )
    def __init__(self):
        self.close_sma = bt.indicators.SMA(self.data.close,
period=self.params.close_period)
        self.volume_sma = bt.indicators.SMA(self.data.volume,
period=self.params.volume_period)

    def next(self):
        if not self.position:
            if self.data.close < self.close_sma and self.data.volume >
self.volume_sma:
                self.buy()
            else:
                if self.data.close > self.close_sma and self.data.volume <
self.volume_sma:
                    self.sell()

cerebro = bt.Cerebro()
cerebro.adddata(data)
strats = cerebro.optstrategy(FirstStrategy,

```

```

        close_period=range(2, 25),
        volume_period=range(2, 25))
cerebro.broker.setcash(startcash)
cerebro.broker.setcommission(commission)
cerebro.addsizer(bt.sizers.FixedSize, stake=stake)
cerebro.addanalyzer(bt.analyzers.SharpeRatio,
    timeframe=bt.TimeFrame.Days,
factor=365)
cerebro.addanalyzer(bt.analyzers.VWR, timeframe=bt.TimeFrame.Days,
    tann=365)
results = cerebro.run()
df_result = pd.DataFrame([{"Close period": result[0].params.close_period,
    "Volume period": result[0].params.volume_period,
    "Sharpe ratio":
result[0].analyzers.sharperatio.get_analysis()['sharperatio'],
    "Variability weighted return (in %)":
result[0].analyzers.vwr.get_analysis()['vwr']
    } for result in results])
display(df_result.sort_values("Sharpe ratio", ascending=False))

```