高频交易下的股票涨跌预测

梅智敏

院 （系）： 计算学部 专 业：软件工程

学　　号： 1183710118 指导教师：张彦航

**2022年5月**



**毕业设计（论文）**

题 目 高频交易下的股票涨跌预测

专 业 软件工程

学 号 1183710118

学 生 梅智敏

指 导 教 师 张彦航

答 辩 日 期

# 摘 要

在数字经济时代，随着数据的规模越来越大，如何压缩和存储是一个基础且重要的课题。然而，传统的数据压缩算法只将重点放在如何提升压缩率，使之接近香农提出的熵的下界，这些算法不适用于全部的场景，特别是对于压缩后的数据有查询需求（如在避免解压全部数据的情况下访问部分数据）时。因此需要一种算法来适应存在着大量对压缩数据有实时查询需求的应用的系统。且随着物联网技术的不断发展、数据的不断增加，基于云的物联网解决方案渐渐无法满足企业的需求。此时边缘计算作为云的扩展，可以加快数据分析的速度，便于企业更快更好地做出决策，但边缘硬件也对算法提出了新的要求，如存储空间更小、不适合做批处理等。因此，本文在边缘硬件环境下实现了基于有损压缩、可对压缩数据进行分析式查询的Plato算法，并通过利用误差向量的正交投影性质，进一步改善了向量乘法运算的error-bound；实现了基于无损压缩、支持随机访问的GD算法，并通过改进预处理时的终止条件，避免了预估空间落入局部最优值。

关键词：时序数据；数据压缩；边缘硬件；误差度量；随机访问

# Abstract

In the era of digital economy, with the increasing scale of data. How to compress and store is a basic and important topic. However, the traditional data compression algorithms only focus on how to improve the compression rate and make is close to the lower bound of Shannon’s entropy.These algorithms are not suitable for all scenarios especially when the compressed data needs to be queried(such as accessing part of the data without decompressing all the data). Therefore, an algorithm is needed to adapt to the system with a large number of real-time query requirements for compressed data. With the continuous development of Internet of things technology and the increasing of data, cloud based Internet of things solutions are gradually unable to meet the needs of enterprises. At this time, as an extension of the cloud, edge computing can speed up the speed of data analysis, which is convenient for enterprises to make decisions faster and better. But edge hardware also puts forward new requirements for the algorithm, such as smaller storage space, not suitable for batch processing and so on. Therefore, in the edge hardware environment, this paper implements the Plato algorithm based on lossy compression, which can query compressed data analytically, and further improves the error bound of vector multiplication by using the orthogonal projection property of error vector; The GD algorithm based on lossless compression and random access is implemented. By improving the termination condition of preprocessing, the prediction space is avoided to fall into the local optimal value.

**Keywords:** time series, data compression, edge hardware, error measurement, random access

# 目 录

[摘 要 I](#_Toc104145642)

[Abstract II](#_Toc104145643)

[目 录 III](#_Toc104145644)

[第1章 绪 论 - 1 -](#_Toc104145645)

[1.1 课题背景及研究的目的和意义 - 1 -](#_Toc104145646)

[1.1.1 高频交易 - 1 -](#_Toc104145647)

[1.1.2 限价订单簿Limit Order Book - 1 -](#_Toc104145648)

[1.1.3 研究目的和意义 - 5 -](#_Toc104145649)

[1.2 国内外研究现状 - 5 -](#_Toc104145650)

[1.2.1 概述 - 5 -](#_Toc104145651)

[1.2.2 基于统计模型的方法介绍 - 6 -](#_Toc104145652)

[1.2.2 基于深度学习模型的方法介绍 - 7 -](#_Toc104145653)

[1.3 该领域存在的挑战及本文的主要研究内容 - 8 -](#_Toc104145654)

[1.3.1 该领域存在的挑战 - 9 -](#_Toc104145655)

[1.3.2 主要研究内容 - 9 -](#_Toc104145656)

[1.4 章节安排 - 10 -](#_Toc104145657)

[第2章 问题形式化及数据集描述 - 12 -](#_Toc104145658)

[2.1 研究问题的形式化 - 12 -](#_Toc104145659)

[2.1.1 MidPrice涨跌预测形式化 - 12 -](#_Toc104145660)

[2.1.2 交易策略形式化 - 13 -](#_Toc104145661)

[2.2 数据集描述 - 13 -](#_Toc104145662)

[2.2.1 时序数据分段 - 13 -](#_Toc104145663)

[2.2.2 估计函数选择与存储 - 14 -](#_Toc104145664)

[2.2.3 误差度量 - 15 -](#_Toc104145665)

[2.3 Plato数据查询 - 15 -](#_Toc104145666)

[2.3.1 误差传播 - 16 -](#_Toc104145667)

[2.3.2 时序数据操作的误差界分析 - 18 -](#_Toc104145668)

[2.3.3 使用正交投影性质改进误差界 - 19 -](#_Toc104145669)

[2.3.4 段不对齐的情况下改进误差界 - 20 -](#_Toc104145670)

[2.3.5 对误差界的进一步改进 - 22 -](#_Toc104145671)

[2.4 本章小结 - 23 -](#_Toc104145672)

[第3章 GD算法的实现与改进 - 24 -](#_Toc104145673)

[3.1 引言 - 24 -](#_Toc104145674)

[3.2 GD数据压缩 - 24 -](#_Toc104145675)

[3.2.1 预处理 - 25 -](#_Toc104145676)

[3.2.2 压缩 - 26 -](#_Toc104145677)

[3.2.3 压缩文件大小分析 - 28 -](#_Toc104145678)

[3.3 GD数据解压缩与随机访问 - 28 -](#_Toc104145679)

[3.3.1 GD数据解压缩 - 29 -](#_Toc104145680)

[3.3.2 GD随机访问及代价分析 - 30 -](#_Toc104145681)

[3.4 本章小结 - 32 -](#_Toc104145682)

[第4章 实验设计与分析 - 33 -](#_Toc104145683)

[4.1 实验环境 - 33 -](#_Toc104145684)

[4.2 Plato算法分析 - 34 -](#_Toc104145685)

[4.2.1 数据集及预处理 - 34 -](#_Toc104145686)

[4.2.2 实验分析 - 34 -](#_Toc104145687)

[4.3 GD算法分析 - 38 -](#_Toc104145688)

[4.3.1 数据集及预处理 - 38 -](#_Toc104145689)

[4.3.2 实验分析 - 38 -](#_Toc104145690)

[4.4 本章小结 - 41 -](#_Toc104145691)

[结 论 - 43 -](#_Toc104145692)

[参考文献 - 44 -](#_Toc104145693)

[哈尔滨工业大学本科毕业设计（论文）原创性声明 - 46 -](#_Toc104145694)

[致 谢 - 47 -](#_Toc104145695)

# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

### 1.1.1 高频交易

在过去的几十年里，新技术正在或者已经改变了股票以及其他金融产品的交易活动。在计算机出现之前，所有的股票交易都是在交易大厅里面人与人之间进行，后台其他人员的处理活动确保了交易的顺利完成。但是，后台处理和实际交易过程中已经被逐渐地自动化。而高频交易就是自动化交易的一个子集，它通常被定义为利用计算机交易算法来提交交易指令和取消交易指令。根据美国证券交易委员会的描述，高频交易具有以下的一些特征：（1）使用高性能的计算机以及复杂的计算机应用程序来产生，路由和执行交易指令。根据最新的行情信息和宏观消息计算产生交易信号是高频交易最核心的功能之一。此外，另有监控系统对交易策略进行实时监控，当异常事件发生时便实时发出警告。（2）将服务器直接架设在交易所，最小化由网络等原因造成的延时在极短的时间内创建和实施买卖交易。（3）采用逐笔数据进行计算。高频交易往往需要以毫秒级的逐笔数据作为计算基础，以消化最新的市场信息，并以此制定交易决策。与人工交易比起来，程序化交易能够更快、有效、理性地决策，这就使得程序化成为高频交易不可或缺的一部分。（4）买卖频繁，持仓时间短。

### 1.1.2 限价订单簿Limit Order Book

传统的金融市场是以报价驱动为基础而运作的，即交易员与交易商（也称“做市商”）在市场中进行沟通。交易商负责维护金融资产的库存量，公布出价和要价，并被要求按其报价进行交易。交易商的存在提供了市场流动性，但是报价驱动市场的透明度相对较低。

而如今，随着电子通信网络和高频交易的快速发展，订单驱动的金融市场成为了主流。它和传统的报价驱动市场最大的区别就在于市场中的交易员可以直接与其他交易员进行交易而没有中间交易商的干预。而订单驱动的金融市场的核心就是以限价订单簿Limit Order Book为中心的双重拍卖机制来促进交易，这一机制已经被全球各大交易所所广泛采用，例如纽约证券交易所（NYSE）、纳斯达克和伦敦证券交易所（LSE）。

在以Limit Order Book为中心的金融市场中，交易者可以提交两大类交易指令：限价指令Limit Order和市场价指令Market Order。前者会要求该交易的成交价格必须好于或者等于所指定的值，如果当前市场中最好的价格仍然不能满足要求，这个交易指令就会继续等待直到市场中出现了满足要求的价格为止，所以此类指令具有“执行不确定性”，即他有可能不会被执行，一直在等待状态。后者则要求该交易立刻以当前市场中最好的价格来完成订单，此类订单一定会被执行，但是不能保证成交价格足够好，所以它具有“价格不确定性”。另外，除了提交这两类指令，交易者也可以取消他们之前所提交但是暂时处于等待状态的Limit Order。

本质上来说，Limit Order Book（也称为 LOB）是当前金融市场行情的一个快照，其基本功能是匹配市场中的买卖双方，它由股票交易所统一管理并呈现给每一位交易者，以帮助交易者们做出交易决策。LOB包含了一组可以代表市场状态及其变化方式的数据且会动态更新，它是当今大多数电子交易所在金融和加密货币市场中使用的机制。图1是LOB的示意图，可以帮助我们直观了解。

**独立统计：**每一支股票的LOB数据分开统计，每一个LOB只反映其对应的那支股票的信息。

**买卖两方**：Ask代表卖方，Bid代表买方。

**价格数量对应关系**：它表明当前市场中有人愿意以多少价格出售/买入多少数量的股票，以Ask方的20.6价格为例，它对应的Volume值为2，代表目前市场中有人愿意以“不低于20.6”的价格出售2支股票。需要注意的是，展示在Limit Order Book中的价格都是Limit Order中所要求的价格，而Limit Order是给出某个价格底线，然后要求成交价“好于或者等于该底线价格”。故Bid方20.5价格对应Volume值为3的含义就是：当前市场中有人愿意以“不高于20.5”的价格买入3支股票。

**买卖双方不同的Level：**仅仅依据交易者的预期价格而无需考虑对应的Volume来对他们提交的“交易需求”进行分级，Level-1是最高级，当市场中的交易发生时，会优先匹配高等级的“交易需求”再到低等级的需求。首先对于Bid方，价格越高则所处的等级越高，故20.5的价格处于Level-1，20.4的价格处于Level-2。接着是Ask方，价格越低则所处的等级越高，20.6的价格处于Level1，20.7的价格处于Level2。

**LOB的更新：**每当市场中有交易员提交了新交易指令或者取消了之前的某个Limit Order，LOB都会更新以呈现出当前市场的真实交易情况给每一位交易者。图2和图3详细展示了两种情况下的LOB更新过程：（1）当有交易者提交了新的交易指令时，LOB会发生更新，图2中展示的是限价为20.55的Limit Ask Order取代20.6成为了新的Best Ask。注意：若有交易者提交新的Market Order时，则会立刻与LOB中记录的Limit Order发生匹配，也会导致LOB的更新。（2）当有交易者取消了尚未执行Limit Order也会引起LOB的更新，图3中展示的是限价为20.6的Limit Ask Order被它的提交者给取消了，所以LOB就会删除这一指令的记录，导致20.7从Level-2 Ask变成了Level-1 Ask（Best Ask）。

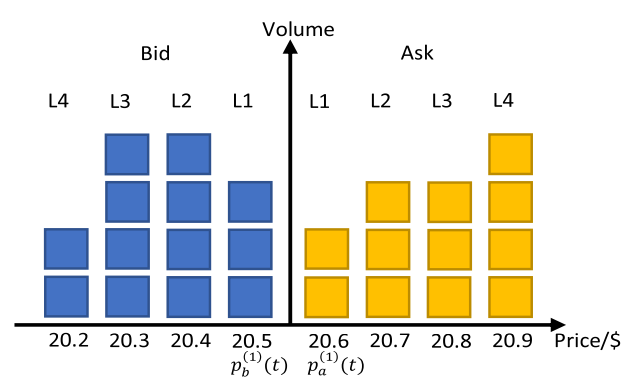


图1 LOB数据示意图

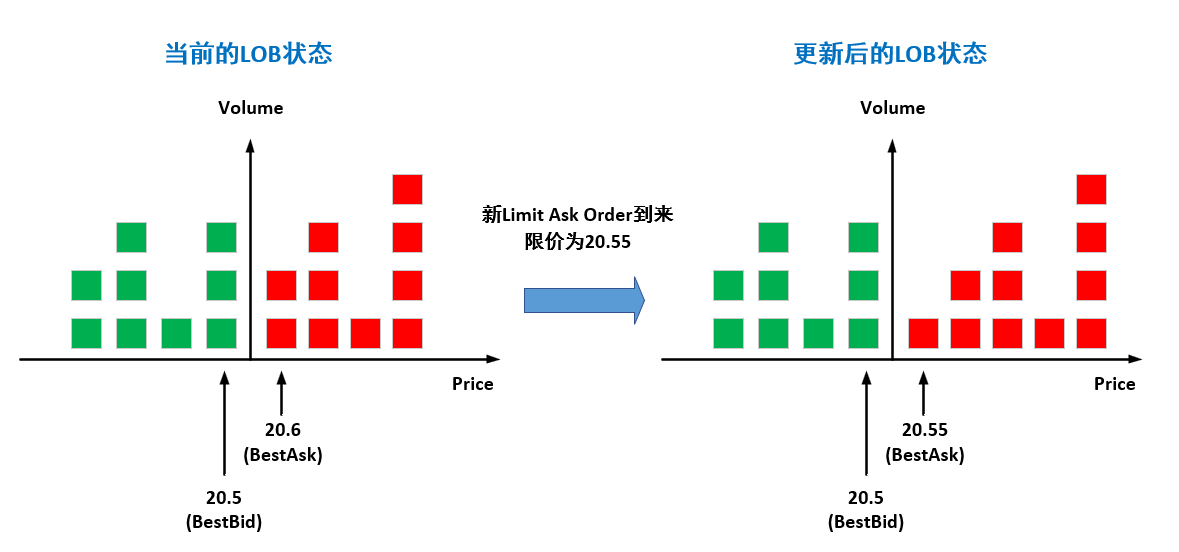


图2 新Limit Order到来导致LOB更新

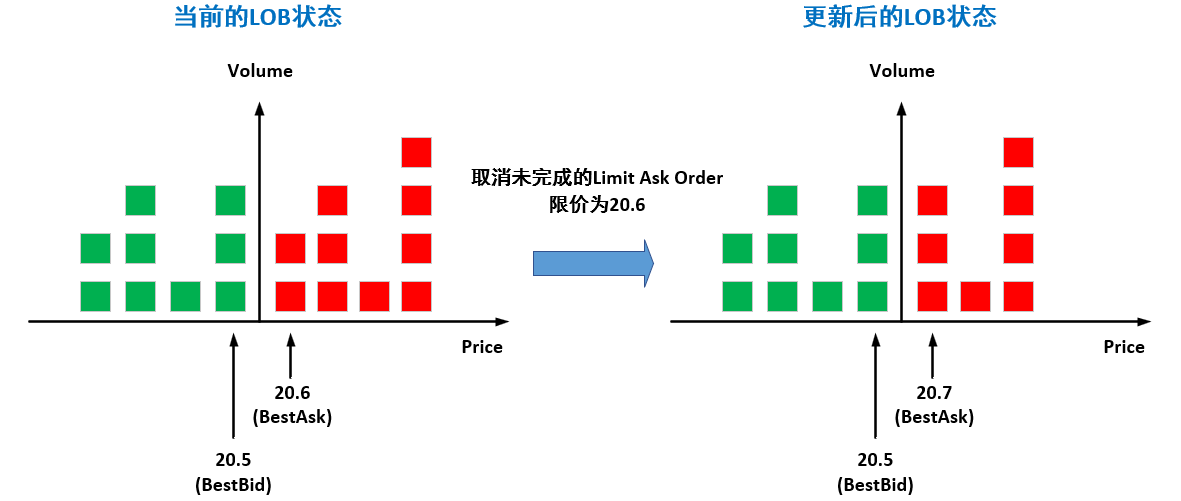


图3 取消未完成的Limit Order导致LOB更新

### 1.1.3 研究目的和意义

在基于LOB匹配机制的金融市场中，**Best Ask**是市场中交易者愿意出售股票的最低价格。也就是说，你可以买到该股票的最低价格。同理，**Best Bid**就是市场中交易者愿意卖出该股票的最高价格。在此引入一个新的指标**Mid Price：**





图4 LOB中Mid-Price的意义

在高频交易市场中，Mid Price是一项重要的指标，它可以表示资产的一般市场价值，所以人们经常使用这一指标来代表资产价值，例如图4中苹果AAPL股票价格其实就是对应LOB中的Mid Price。

因此，对于量化金融研究者而言，如何准确预测短期内LOB中Mid Price的涨跌情况成为了近年来的热点问题。目前的高频交易研究主要包含两大类：价格预测和优化策略，而价格预测也正是优化策略的基础。因此借助LOB进行短期Mid Price变化预测可以有效指导交易者们合理优化自己的交易策略以获得最大收益，具有重要的实际意义。

## 1.2 国内外研究现状

### 1.2.1 概述

已有的方法主要可以分为两大类：统计模型方法以及数据驱动的机器学习方法。其中前者是传统的主流方法，优点是不需要利用大量的数据来训练，复现较为简单；缺点是这类方法经常会对数据的分布进行一些和实际情况不相符合的假设，从而导致应用到实际的高随机性的股票数据时，预测效果往往不好。例如[1]中所提出的VAR模型以及[2]中提出的ARIMA模型。而今年来随着深度学习的兴起，研究者们更倾向于使用数据驱动的深度学习方法来完成高频交易下短期股票涨跌趋势预测任务，本课题也挑选了部分已有的方法进行了复现。

### 1.2.2 基于统计模型的方法介绍

试图利用统计模型或者随机模型来捕获Limit Order Book中的动态特征，并在此基础上准确预测短期内股票价格涨跌趋势是很多统计学派研究者们所希望做到的。

例如，Gourieroux（1999年）[3]、Bouchaud（2002年）[4]和Smith（2003年）[5]通过套用多种统计模型进行试验挖掘出了Limit Order Book中20多种动态特征，并给出广泛统计特征列表。然而，他们的工作仍然不能将特征统一到单个统计模型中加以使用。可喜的是，Cont和de Larrard在2011年的工作[6]中，填补了上述空白，他们提出使用马尔可夫模型来模拟以Limit Order Book为基础的金融市场运作过程，该模型捕捉了Market Order以及Limit Order的主要特征并在此基础上模拟这些特征对价格走势的影响。但是他们的模型仅仅考虑了LOB中Level-1的特征，即大部分剩余特征并不在考虑范围内。值得一提的是，相同的作者Cont在他的另一个工作[7]中确实考虑到了多个Level的Price以及对应的Volume信息，他假定所有即将到来的市场事件（Limit Order，Market Order以及Cancel Order）都遵循独立的泊松过程，以此来预测未来的MidPrice。受到Cont和de Larrard使用马尔可夫模型成功的启发，越来越多的学者开始研究具有随机驱动项的类马尔可夫模型，如向量自回归模型（VAR）[1]和自回归综合移动平均模型（ARIMA）[2]。但是这些模型为了避免过多的参数空间，通常依赖于人工创建的数据特征，但是以此类特征为基础的统计模型大多不具备泛用性，它们往往在某几支股票上表现很好但是在其他股票上表现很差。

总的来说，在2015年之前，使用传统的统计模型来理解Limit Order Book动态过程的研究工作还较多，近年来则是大大减少了。其根本原因就在于这类方法要么大量使用一些人工创建的特征，要么对数据或者市场中事件的分布进行一些和实际情况不相符合的假设，从而导致模型难以应用到实际的高随机性的股票数据。

### 1.2.2 基于深度学习模型的方法介绍

近年来随着深度学习的快速发展以及传统统计模型难以有效捕获LOB的动态特征，越来越多的深度学习模型被应用到金融数据中以尝试利用Limit Order Book中的多维市场交易信息来预测短期内金融资产（股票、证券以及外汇等）的价格走势。其中MLP和SVM是两类常用的分类模型，例如在[8]中，作者训练了一个SVM模型来预测未来30s内日经225指数的价格涨跌走势。在[9]中，作者使用12个维度的特征向量作为输入，来预测韩国综合股价指数的日内价格走势。并且在两个不同大小的预测窗口下（一个为长期，另一个为短期）对MLP和SVM的性能进行了比较。在[10]中，作者通过一个预测期货未来价值的任务来对MLP，SVM和RBF-NNs进行了实验比较。

除了在不同的金融数据集中比较分类器的性能外，也有很多学者将研究重心放在如何从Limit Order Book中提取合适的特征来帮助后续预测任务。由于金融数据的高频高随机特性，没有预处理或者特征提取的话会给后续模型训练带来极大困难。为此，[11]中的作者在模型训练之前使用主成分分析PCA来对源数据进行降维从而使得相同的模型性能得到了提升。但是，这种将特征提取方法是静态的预处理步骤，没有随着模型训练过程实时优化以最大限度地提升模型的整体性能。为了弥补这一缺陷，一些学者提出将特征提取功能设计成模型中的某个单独的网络层，这样在模型训练过程中就可以实时利用反向传播来更新自身参数以达到优化的目的。相关的代表工作有[12]中的BoF模型以及[13]中的CNN模型，在[12]中，作者在公开数据集FI-2010上进行了实验，这是基于Limit Order Book进行股票价格预测的标准benchmark数据集，包含连续 10 天从纳斯达克北欧股票市场中提取的五只股票的LOB的标准化数据表示，约4,000,000 个样本点。在[13]中，作者使用卷积神经网络 (CNN) 来进行特征提取，创新性地将该方法应用到高频股票数据中。该方法使用来自金融交易所的非公开大规模、高频LOB数据作为输入来预测股票的价格走势，并且在实验中与其他不做特征提取的方法（MLP）进行比较。这类工作表明了从大量数据中提取代表性特征对于理解Limit Order Book的动态过程具有重要价值。

还有一类模型近年来也在金融数据领域备受关注，那便是LSTM[14]。该模型最初是为了解决循环神经网络RNN对于长时依赖的消失梯度问题[15]，现在已经广泛应用于NLP领域，也常常在seq2seq [16]模型架构中充当encoder或者decoder的编码器。近年来越来越多的研究者使用LSTM来抓取股票历史数据之间的“时序依赖信息”从而帮助预测MidPrice走势。代表工作有[17]，作者们使用1000只股票的Limit Order Book数据来测试他们设计的四层LSTM模型。结果显示，随着时间的推移，样本外预测精度并未明显下降，这体现了在金融数据中LSTM具有不错的潜力。

总的来说，近年来深度学习的研究愈发火热，基于DL模型来理解LOB动态过程的研究大量涌现。其主要围绕三大主题：如何从LOB中提取合适的特征，如何从时间维度捕获LOB数据之间的依赖关系以及如何让模型在不同大小的预测范围下仍保持良好的表现。

## 1.3 该领域存在的挑战及本文的主要研究内容

### 1.3.1 该领域存在的挑战

综合考虑前面的领域内研究调查，我将该领域内的挑战总结为三点：

(1) 对于统计模型，难以在足够小的参数空间和合理的分布假设之间做到平衡，往往是顾此失彼。

(2) 对于深度学习模型，如何从LOB中提取合适的特征，如何从时间维度捕获LOB数据之间的依赖关系以及如何让模型在不同大小的预测范围下仍保持良好的表现都是当前面临的难点。

(3) 需要充分的实验来佐证预测模型的优良性能，主要包括两方面的测试：数据集上的模型指标测试以及回测收益测试。前者是利用一些评价指标如Accuracy、Precision、Recall以及F1值来比较不同模型在同一个数据集上的表现好坏；后者则是依据预测模型的结果设计一个交易策略，然后在历史数据上模拟执行此交易策略，根据最终得到的收益或者损失Profit&Loss（PnL）来从另一个角度衡量不同策略的好坏。

### 1.3.2 主要研究内容

本次毕设的主要研究内容为在高频交易的应用场景下借助Limit Order Book数据来对股票的涨跌走势进行预测，同时根据预测结果制定可以获利的交易策略。研究主要分为以下几个阶段：

(1) 首先需要广泛阅读文献，了解目前的研究者们主要采用统计模型还是深度学习模型、提取了Limit Order Book中的哪些特征、设计模型的依据、一般在哪些数据集上进行测试。

(2) 其次，需要从已有文献中获得启发，在贴合应用场景的基础上设计出新的分类模型用于预测短期内MidPrice的涨跌走势，模型尽量不要对数据的分布做出不合理的假设。

(3) 另外，从两个大方向来优化自己的模型：模型设计与特征提取。模型设计这块可以尝试引入seq2seq架构让模型可以同时输出多个预测范围内的价格走势结果；特征提取这块可以先尝试手动设计几个特征，然后利用和标签之间的相关性检验来证明特征的有效性。

(4) 最后，需要进行充分的实验来评估设计的模型，实验分为两大块：评价指标测试和回测盈亏。每一块都包含3大阶段：baseline方法的表现、初步模型的表现以及优化后模型的表现。需要注意的是实验中需要对预测范围T取不同值进行多次验证。

## 1.4 章节安排

本论文共5章，各章内容概述如下：

第1章为绪论，首先介绍了课题的背景以及研究的目的和意义。从整体上阐述此毕设存在的重要性。然后，介绍了国内外在时序数据压缩查询方向的研究现状，先是介绍了普遍用于压缩算法的压缩技术，然后对该研究方向的算法进行分类并一一介绍。最后，阐述了在该领域存在的挑战以及本次毕业设计的主要研究内容。

第1章为绪论，主要分为三个模块。首先介绍了课题的应用场景以及现实意义，详细描述了现代以Limit Order Book作为匹配机制的金融市场运作方式，指出基于LOB的短期MidPrice涨跌趋势预测是一大热点；其次对领域内已有的研究方法撰写了文献综述，传统的基于统计模型的方法和新兴的基于深度学习模型的方法都有介绍；最后从前人的方法中得到启发，确定了本课题模型优化方向以及需要的两大类验证实验。

第2章为问题形式化及数据集描述，该章分为两节。第一节对本课题研究的问题进行形式化并统一相关符号以便于论文后续表达；第二节对本课题所使用的公开数据集FI-2010进行详细描述，包括特征集以及标签集。

第3章为初步模型的设计与实现，该章分为四节。第一节介绍深度学习模型中的CNN和LSTM，本课题的初步模型主要用到了这两大模型；第二节详细描述初步模型的设计思路与模型结构；第三节结合数据集和模型结构给出数据流向图，详细描述了从模型输入到模型输出的向量维度和形状变化情况；第四节基于初步模型设计了一个交易策略，介绍了设计思路与实现方法。

第4章为对初步模型的改进思路与实现，该章分为四节。第一节阐述从模型角度的改进，即引入seq2seq结构和注意力机制的动机与设计思路；第二节介绍从特征提取角度的改进，即引入两个新特征OI和OIR的原因、设计思路和有效性验证；第三节综合前面两节的内容对初步模型加以优化，给出实现细节和优化后的模型；第四节描述基于改进后模型的交易策略设计思路及实现细节。

第5章为实验设计与实现，该章分为两节，每一节都包含模型指标测试和回测盈亏两部分实验，模型指标测试会比较模型的Accuracy、Precision、Recall和F1值而回测盈亏会比较基于不同模型的交易策略在历史股票数据上的盈亏情况。第一节首先介绍实验设计，然后将初步模型和文献综述中提到的一些已有深度学习方法进行比较；第二节将初步模型和改进后的模型进行了比较，具体行文结构与第一节类似。

在最后的结论中，本论文对完成的工作进行了总结：阐述了初步模型和改进后模型的设计思路中的创新点所在；对两阶段实验得到的结果进行了综合分析与评价；同时针对高频交易中的股票涨跌预测问题对未来进行展望，提出了几种可能的研究方法。

# 第2章 问题形式化及数据集描述

## 2.1 研究问题的形式化

第一章中已经对本课题所要研究的问题进行了简略描述，此处给出更为准确的形式化表达。分为两个阶段，第一阶段是预测MidPrice的涨跌趋势，第二阶段是基于涨跌预测的交易策略。

### 2.1.1 MidPrice涨跌预测形式化

根据第一章中对LOB的详细描述，我们可以将时刻的LOB状态用向量表示：



其中不同的代表LOB中不同的Level，每一个Level包含买方价格、买方价格对应的量、卖方价格、卖方价格对应的量这4个维度的信息，分别用、、、表示，示意图如下：

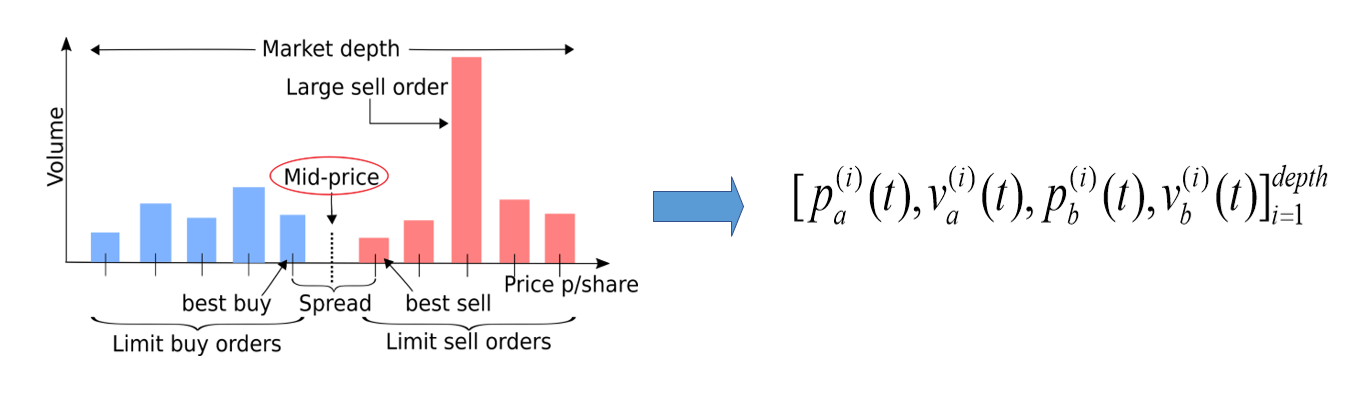


图2-1 LOB状态表示

每当市场中交易者提交新Limit Order、提交新Market Order或者取消尚未执行的Limit Order，作为市场买卖行情快照的LOB就会自动更新一次。关注每次更新的LOB状态，并用相同的向量化表示它们，就可以得到一串LOB状态序列。每一个都对应着一个MidPrice，记作：



根据第一章的描述，可以衡量该时刻的股票价值，那么我们就可以通过观察MidPrice序列的变化情况来判断股票价值的涨跌。但是由于股票数据具有很强的随机性，如果我们仅仅比较和来判断股票价值的涨跌，那么最终得到的结果将会同样随机而难以找到规律。因此，在很多已有的研究中[12][18]，人们引入了一定时间跨度内的增长率来表征股票价值的涨跌情况，具体公式如下：





其中代表未来个样本点的平均MidPrice，而代表过去个样本点以及当前样本点的平均MidPrice，接下来引入增长率的概念：



表示未来时间范围内的股票平均价值相较于过去时间范围内的股票平均价值涨幅。另外，股票价值可以分为涨、跌、保持相对平稳这三大类，于是可以自定义标签如下：



其中，是判断阈值，一般取0.001到0.002之间。在本课题中，沿用前人在[13]中的取值0.002。

前人的研究[17]发现可以使用LSTM来捕获之间的时序依赖关系从而帮助预测，我们可以沿用这一思路，但是观察对象是能提供更多信息的LOB状态序列。于是我们将这个预测问题最终形式化如下图：

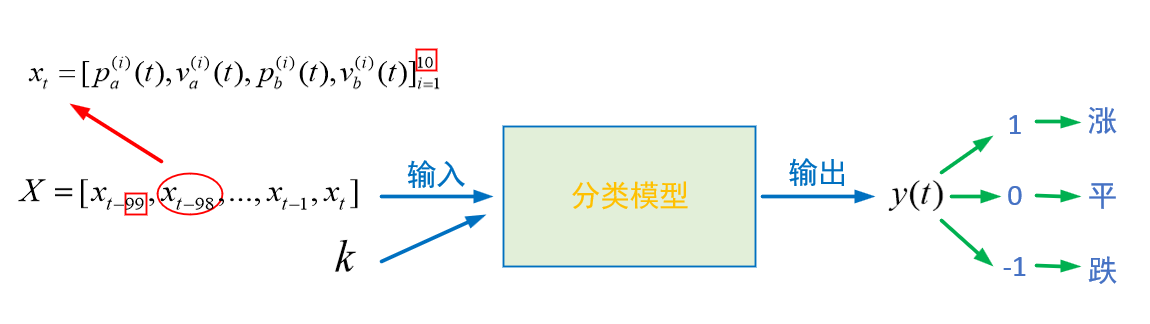


图2-2 预测问题形式化

我们观察包含当前在内的共计100个序列，为了控制输入向量的维度，每个使用前10个Level的信息，即每个包含40个维度的特征。分类模型的输入的维度是100\*40，另外需要输入预测范围，经过分类模型处理之后输出对的预测结果：涨或跌或平。

### 2.1.2 交易策略形式化

依据分类模型的预测结果可以设计出一种简易的交易策略：若意味着短期内股票价格上涨，我们需要尽可能买入；若意味着短期内股票价格下跌，我们需要尽可能卖出；若意味着短期内股票价格保持平稳，我们就不做任何操作。由此可以定义动作,可以取值，形式化表达如下图。假设预测模型在处的输出，则，我们需要向交易所提交Market Buy Order以尽快满仓买入。为了方便后续的回测盈亏实验，我们直接以BestAsk作为成交价；同时考虑到模型输出和提交指令的时间耗费，当我们的Market Buy Order到达交易所时，LOB可能已经更新了几次，所以成交价采用而不是，为延迟因子，在本课题中取。同理，若预测模型在处的输出，则，我们需要向交易所提交Market Sell Order以尽快空仓，成交价使用。

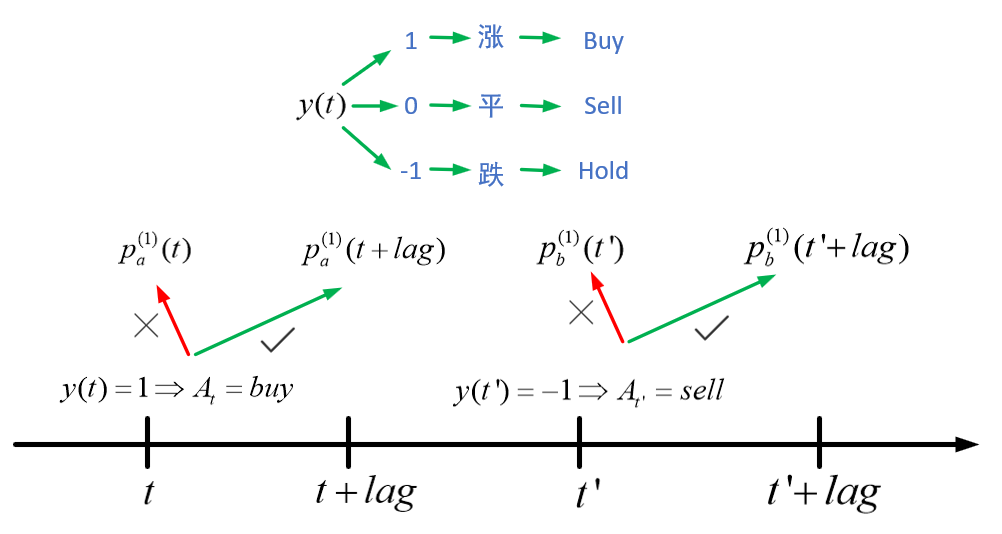


图2-3 交易策略形式化

## 2.2 数据集描述

### 2.2.1 来源及包含的股票

### 2.2.2 特征集

### 2.2.3 标签集

# 第3章 初步模型的设计与实现

## 3.1 引言

支持访问式查询的时序数据压缩算法使压缩数据可应对访问式的查询，我们选定该类算法中的GD算法[15]进行实现并优化。传统的数据去重算法会将仅有一点差别的块视作不同块，显然这浪费了块之间潜在的相似性。GD算法利用了推广的去重思想，通过提取块之间相同的部分为base并进行去重，不同的部分为deviation按原数据存储，更好地利用了不同块之间的相似性，并以非常低的代价支持随机访问。

在本章中，我们主要阐述了实现GD算法的过程和对其进行的优化。图3-1说明了GD算法的运行流程，在压缩阶段，算法对时序数据进行分块，并通过转换函数对块进行重排列、提取base和deviation，最终base被存储在字典中，deviation存储在压缩后的数据中；在查询阶段，通过读取字典和参数可对原数据进行随机访问。

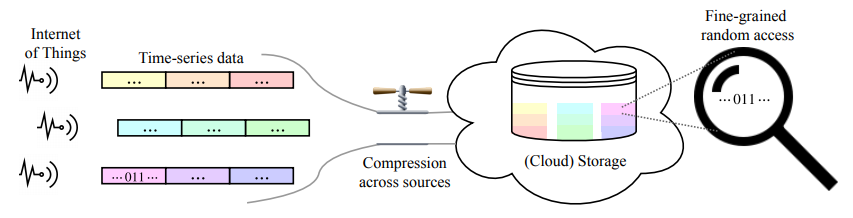


图3-1 GD算法运行流程

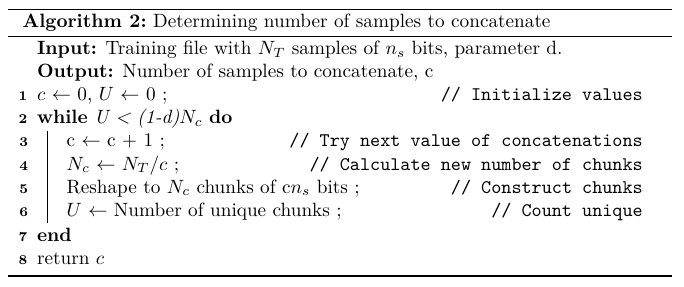
## 3.2 GD数据压缩

GD算法数据压缩分为两个阶段，预处理阶段将待压缩数据的开头一部分数据作为训练数据，从而推断出压缩算法的关键参数。压缩阶段通过参数构建压缩器，对文件以流处理的方式进行一个一个地压缩。

### 3.2.1 预处理

预处理阶段通过利用开头一部分的压缩数据作为训练数据，以确定压缩过程的关键参数，这部分数据可以是文件流中的第一个文件，且必须具有代表性，即能反应数据的整体结构，首先设定每个数据由个比特位表示。

该阶段首先要计算的参数是构成数据块所需要数据的个数c，最佳的c值取决于相邻数据之间的相关性，如果设置的较小会减少编码base ID所需的位数，但可能对块相似性的利用不够充分；如果设置的较大会降低base相同的可能性，因为需要在更多的位数上相同，合适的c值应该从数据中推理出来，该算法的伪代码如下，随着算法的迭代，构成块的数据越来越多，直到几乎每个块都不同（允许比例为d的块相同，实验中该值取0.01），算法终止时c的值即每个块中数据的个数，每个块包含个比特位。

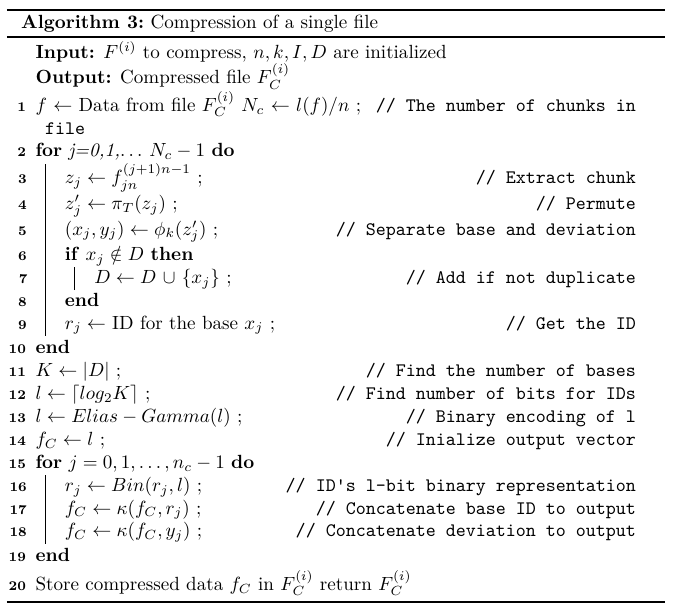


下一个要确定的参数是块中构成deviation的位数，为了使相同的base数量尽可能多，我们想让deviation包含和块中其他位关联最小的位。举个例子，10001、10010、10000、10011这四个块中，显然最后两位与其他位关系最小，因此最适合做deviation。这里通过估计位之间的互信息来表示“关联”的大小，但估计互信息通常是很困难的，因此采用皮尔森相关系数来替代，根据[16]的研究，皮尔森相关系数携带了和互信息相同的信息。将每个bit到其他bits的相关系数平均值按降序排列，越靠后的bit越适合作为deviation，为了计算最适合的deviation位数，不考虑参数的存储先粗略地表示出压缩后文件的大小：

其中K表示不同base的数量，N表示块的数量，首先令k=n，然后每次迭代令k减一，直到即估计大小达到了局部最优值。在实验过程中，我发现算法有时会过早到达局部最优值，导致最终压缩效果很差，因此我将该迭代算法的结束条件优化为，避免了很小的扰动就使算法终止。最终将作为deviation的n-k个bit的索引存入*I*中，并在原数据块中将其移动到块的最后面。

### 3.2.2 压缩

该阶段对数据进行压缩，压缩时文件可以以文件流的形式，即一个一个地到达，如果文件不能被划分为整数个块，最后一块使用零填充。算法的伪代码如下，第一趟扫描通过维护base字典，对文件中的每一个块分别进行重排列、提取base、查询字典、分配base ID；第二趟扫描将base ID和deviation拼接存储，建立压缩后的块。第一趟处理后，会计算编码base ID所需的位数，其中为字典的当前大小，并将其用Elias-gamma编码存储在文件头部。



最终的存储格式见图3-2，原文件分块后进行压缩，文件流经压缩后生成三种文件：压缩器参数、字典和压缩数据。压缩器参数存储单个数据所占比特数，每个数据块包含的数据个数和重排列索引；字典包括所有数据块中不同的base；压缩数据包含指向字典中对应条目的base ID，和块之间不同的部分deviation。

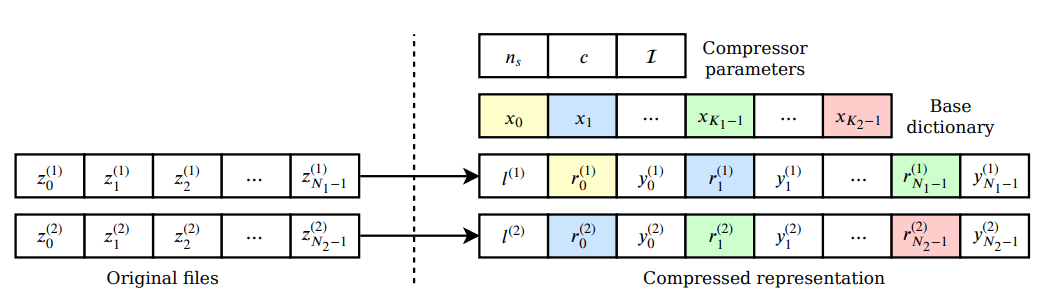


图3-2 原文件格式与压缩后的文件格式

### 3.2.3 压缩文件大小分析

我们使用Elias-gamma编码来编码整数，Elias-gamma编码是一种简单的前缀编码方式，任何整数a都可以使用位比特表示，需要存储的整数包括每个数据所占比特数，每个数据块的数据个数和重排列字典中的索引，需要的总比特数为

压缩后的单个文件大小表示为，该值与原文件中的数据块个数和编码base ID所需要的比特位数有关，所以压缩后的文件大小可表示为

其中，第一项为编码的代价，第二项为编码原文件每个数据块的代价，其中包括长度为的base ID和长度为的deviation。

存储base的字典的代价也要计入，可表示为

其中为字典的大小

最终，压缩后的数据总大小为

对于一般的数据来说，(3-2)的开销都可以忽略不计，真正的挑战在于选择合适的参数平衡(3-3)和(3-4)的开销，这也是预处理阶段的目标。

## 3.3 GD数据解压缩与随机访问

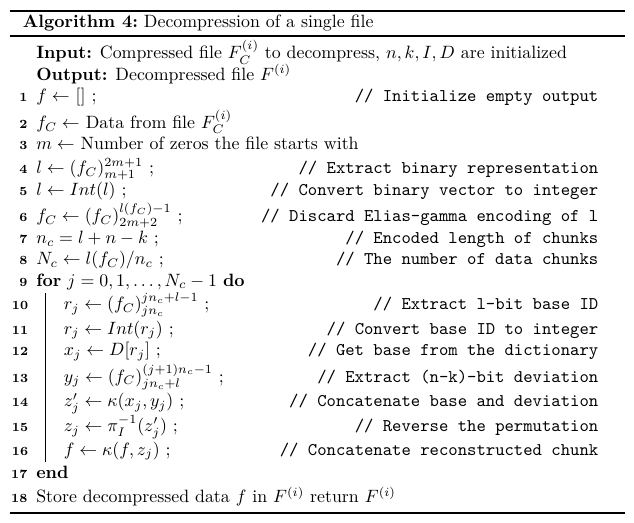
GD算法解压缩可对压缩数据进行完全的解压缩，随机访问是推广的去重压缩方法提供的特性，允许对原数据的任意比特位进行低代价的随机访问。

### 3.3.1 GD数据解压缩

GD数据解压缩分为两个阶段，首先要恢复压缩器状态，然后对文件进行解压缩。

恢复压缩器状态需要先读取预处理阶段计算的参数和，然后可计算出数据块的大小为个比特，base的长度为，最终读取字典。由于字典的组织形式是可预知的，字典中第j个连续k比特块对应的base ID为j-1，所以后续的解压缩可以直接根据base ID找到对应字典中的base。

解压缩阶段的算法伪代码如下，压缩文件开头的Elias-gamma编码首先被恢复，得到base的位数，随后每读取到位作为base ID，紧随其后的位作为deviation，由于已被恢复，可通过中存储的重排列索引将deviation恢复至原位，从而得到原来的数据块。



### 3.3.2 GD随机访问及代价分析

推广的去重压缩方法允许对原数据的低代价随机访问。首先必须读取一部分数据以恢复压缩器状态，即预处理阶段的参数和，并计算出和，恢复这些数据需要访问的比特数为，即公式(3-2)的值，该部分的代价是独立于所存储数据的大小的，即所有对数据的访问都存在的代价。下面将按照数据访问方式分别计算代价。

访问单个数据块时，令我们要访问的块为，即第i个文件的第j块，首先要读取文件头部的，即编码base ID所需的位数，然后计算出该块的起始位置为

结束位置为

通过块中前位，即base ID，可找到字典中对应原base的位置，恢复出原块后再根据中存储的重排列字典进行位置恢复，即得到重排列前的块，总访问比特数为

仅仅比未压缩的数据多访问了位，如果参数已被恢复，那么仅需位。

访问某个比特位的代价取决于该位是base还是deviation，假设我们要访问第i个文件的第a个bit，在该文件中，该bit应定位到块，在该块中，a应位于位置a mod n。

**访问deviation bit** 如果，那么该bit属于deviation，如果是的第j个元素，则该bit是deviation的第j个元素，该位置为，总访问bit数为：

**访问base bit** 如果，那么该bit属于base，首先读取块前的位以确定base ID ，于是可确定要访问的base在base字典中起始位置为，结束位置为，还需知道有多少个deviation bit原来在前，现在被重排列到了块的后面，该值可表示为

其中表示指示函数，当内部表达式为true时取1，false时取0，所以要访问的bit在字典中位于，总访问bit数为：

## 3.4 本章小结

在本章节中，我们对GD算法的数据压缩、解压缩、随机访问过程进行了分析。GD算法使用推广的数据去重方法，将数据块划分为base与deviation，其中对base使用传统的数据去重方法进行去重，对deviation按原数据存储，且考虑到deviation不总在数据块的最后面，通过皮尔森相关系数估计互信息，将deviation移动至数据块的最后面。推广去重解决了相似块无法被去重的问题，且由于去重算法的特性，可通过压缩参数和数据字典恢复压缩器状态，从而对原数据进行低代价的随机访问，我们对随机访问的代价进行了分析，分析得出在压缩器状态已被恢复时，随机访问的代价非常低。我们还对该算法预处理阶段的终止条件进行了改进，避免了很小的扰动就使算法终止。

# 第4章 实验设计与分析

## 4.1 实验环境

本次实验目的是在边缘硬件上实现时序数据压缩与查询算法，因此使用华为的面向广泛边缘应用场景的轻量边缘设备Atlas 500作为算法部署环境，Atlas 500的系统架构图如图4-1所示，处理器为华为自研海思Hi3559A，在该处理器上运行了一个基于Linux内核的定制操作系统Euler，用于管理硬件资源。由于Euler基于Linux内核，因此为了保证算法的可移植性，采用ubuntu 18.04作为开发环境，c++作为算法实现语言。

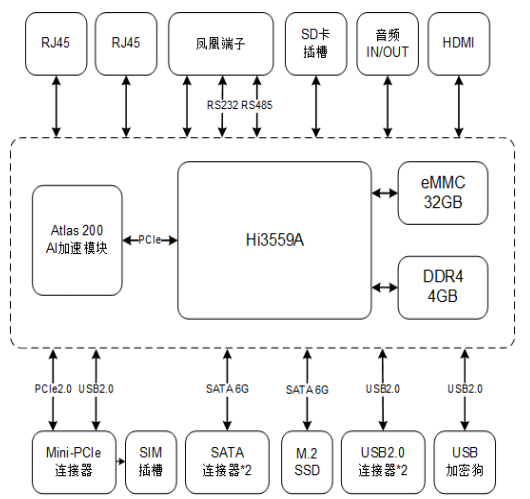


图4-1 Atlas 500系统架构

就算法本身来讲，运行在边缘设备和普通计算机上差别不大，但边缘设备在应用场景上更接近数据的源头，对数据的处理更加及时，因此在边缘设备上的实验设置与在开发环境的实验设置基本相同。同时考虑到边缘设备有限的存储容量和未来要面对的大规模数据的压缩需求，算法要尽可能地具有实时性，本次毕设选定的两个算法均具有离线压缩、在线查询的能力，符合边缘硬件对算法提出的新要求。

## 4.2 Plato算法分析

### 4.2.1 数据集及预处理

在该算法中，我们使用了两种真实数据集：比特币成交记录数据集(HB)和空气质量记录数据集(HA)，表4-1总结了这两种数据集的统计信息。

表4-1 Plato数据集统计信息

|  |  |  |  |
| --- | --- | --- | --- |
|  | 平均数据点个数 | 时序数据数量 | 单位 |
| HB | 4,860,000 | 8 | 分钟 |
| HA | 1,587,258 | 11 | 分钟 |

HB数据集包含了从2012年到2018年每分钟的数字货币成交价格，以逗号为分隔符，csv格式存储，数据中有缺失值。对该数据集进行处理时，我们使用前一个非缺失值去填充当前的缺失值，并将每一个时序数据单独导出，不包含列名。

HA数据集包含了从2011.9.10到2014.9.10圣地亚哥市11个不同的空气质量指标如气压、气温、风向等，数据中无缺失值。对该数据集进行处理时，直接将每一个时序数据单独导出，不包含列名。

### 4.2.2 实验分析

本次实验使用已被证明效果更好的滑动窗口算法[10]作为时序数据分段算法，使用具有正交投影性质，可对error-bound进行进一步优化的一阶多项式作为时序数据段拟合函数。

我们首先通过在HA数据集上执行查询，评估了真实误差（估计值与真实值的误差）和误差界的大小及比例关系，结果见表4-2。如表所示，进行分析的操作包括单时序数据操作：均值、标准差，Plato对该类型的查询给出了接近100%准确的结果；对于多时序数据操作相关系数，Plato给出了相差一个量级的error-bound，对于自相关，Plato给出了同量级的error-bound。

表4-2 时序数据分析操作的真实误差与误差界

|  |  |  |  |
| --- | --- | --- | --- |
| 时序数据分析操作 | 真实误差(True Error) | 误差界(Plato) |  |
| 均值 | 0.00 | 0.00 | 0.00 |
| 标准差 | 0.0221474700 | 0.0221474700 | 0.00 |
| 相关系数 | 0.0041128962 | 0.0626709724 | 14.2376740548 |
| 自相关 | 0.0157846300 | 0.0620285434 | 2.9296799036 |

Plato算法可通过控制分段阶段的误差参数，从而控制压缩文件的压缩率。由于后续实验部分会通过调整参数将压缩文件大小调整到很低，此处压缩率定义采用原文件大小除以压缩文件大小，即压缩率为100表示压缩文件大小为原文件的1/100。为了验证不同压缩率对真实误差与误差界的影响，我们固定查询为相关系数，通过调整参数控制压缩率得到了HB数据集上查询的真实误差与误差界，结果见图4-1。总体来看真实误差在误差界的范围内，但随着压缩率的提高，误差界对真实误差的限制逐渐放松，即二者的差距逐渐增加。误差界在压缩率为10和20时比真实误差高一个量级，在压缩率为50时比真实误差高两个量级，但二者的值作为相关系数的误差都非常小（在水平），因此相比于真实结果0.9999983084几乎可以忽略不计。

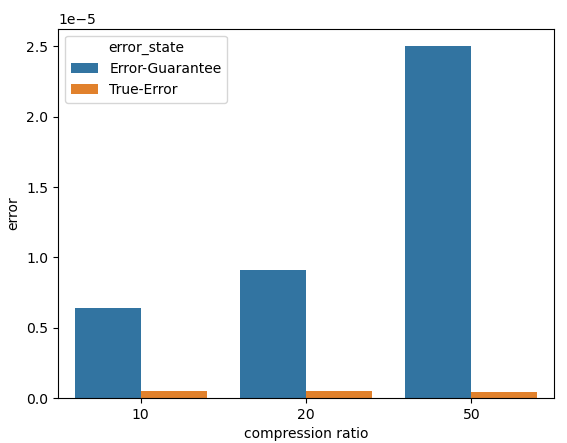
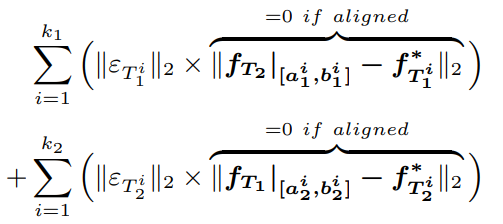


图4-1 HB数据集上的true error与error bound

作者在对段不对齐时的进行估计时使用正交投影性质将公式2-4①优化为了



该式只在与均值相近时才会达到优化的效果，因为此时与同号且同量级，相减后才会在原值基础上大幅降低。但在与相差几个量级，或者与是异号的时序数据时，与相减会导致原值基本不降低，甚至可能不降反增。下面的实验说明了这点，图4-2展示了在HB数据集上两个均值分别为3077.54134和3077.508174的时序数据的相关系数查询结果误差界，此时时序数据均值几乎相等，优化后的误差界没有显著的降低。图4-2展示了在HA数据集上均值为916.8301267和61.85144078的两个时序数据的相关系数查询结果误差界，此时时序数据误差相差一个数量级，优化后的误差界显著降低。

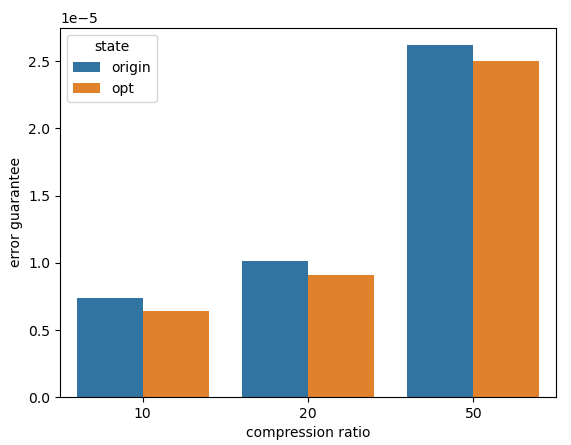


图4-2 HB数据集上的优化前后误差界

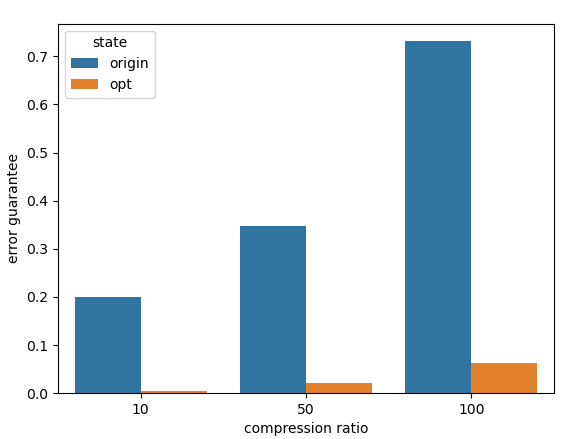


图4-2 HA数据集上的优化前后误差界

## 4.3 GD算法分析

### 4.3.1 数据集及预处理

在该算法中，我们使用了三种数据集，表4-2总结了这三种数据集的信息。

表4-2 GD算法数据集介绍

|  |  |  |
| --- | --- | --- |
| 数据集名称 | 描述 | 大小 |
| ECG | 麻省理工学院心律失常信息数据库 | 93.6MB |
| T-DK | 丹麦8个地点的日最高气温 | 2.1MB |
| Elec | 10个家庭的用电量情况 | 4.0MB |

ECG数据集记录了麻省理工学院收集的心律失常的心电图数据，数据集中包含三种数据：.atr：注释数据，记录了相应心电图信息的种类；.dat：数据文件，按二进制存储；.hea：头文件，按ASCII码存储，包含一些基本的常量。本次实验对.dat文件进行压缩。

T-DK数据集记录了丹麦8个地点的日最高气温，以文本形式存储整数，包含21行的文件头。对该数据集进行预处理时，首先跳过前21行的文件头，然后将剩余数据转换为二进制。

Elec数据集记录了10个家庭的用电量情况，以文本形式存储浮点数。对该数据集进行预处理时，读取所有的浮点数并转化为二进制形式存储。

### 4.3.2 实验分析

为了比较基于推广去重方法的GD算法与传统的压缩算法的压缩率，我们选定两种传统算法与GD算法进行压缩性能上的比较：GZIP算法是DEFLATE算法的实现，应用了LZ77编码和哈夫曼编码的思想；7-zip算法使用了LZMA算法进行压缩，LZMA算法是LZ77算法的扩展。由于无损压缩保证了信息的无损失，其压缩效果通常比有损压缩差，所以后续实验部分的压缩率定义为压缩文件的大小除以原文件的大小，即压缩率为0.1表示压缩文件大小是原文件的1/10。

三种压缩算法对三种数据集的压缩率见图4-3，GD算法的运行参数在柱状图中标明。对于前两个数据集，GD算法的压缩率略低于GZIP算法，高于7z算法；对于最后一个数据集，GD算法的压缩率显著高于GZIP算法和7z算法，可以看到此时deviation的比特数退化为0，即推广的去重算法退化为传统的去重算法，导致了压缩效果一般。

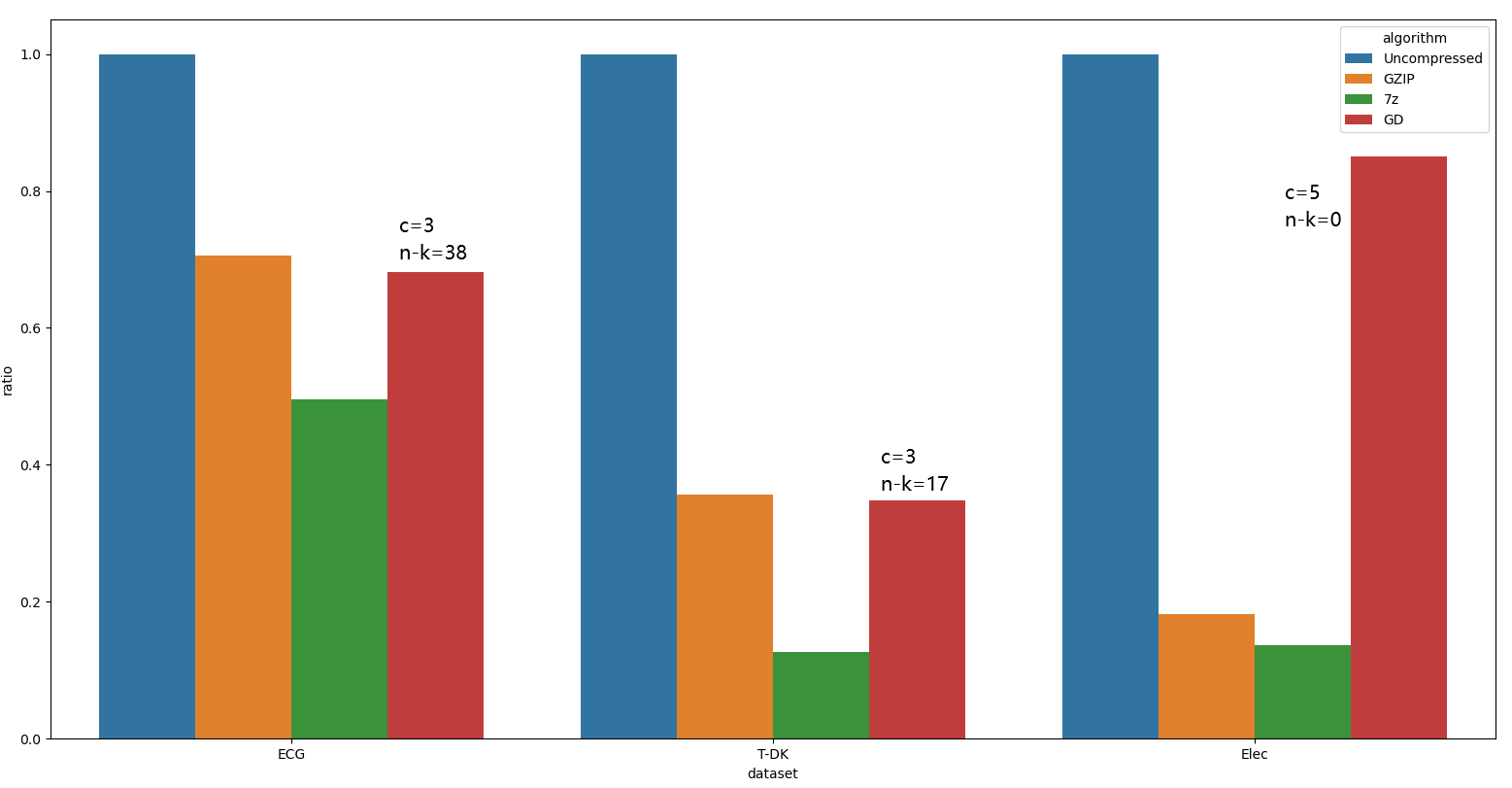


图4-3 三种算法对三种数据集的压缩效果

为了验证预处理阶段的参数选择对最终压缩率的影响，我们通过手动改变参数c和n-k的值，计算了GD算法在T-DK数据集上的压缩率，结果见图4-4。启发式算法选择的参数c=3，(n-k)/n=0.2，可见此时的参数非常接近最小值，说明启发式算法的结果是具有可信度的。

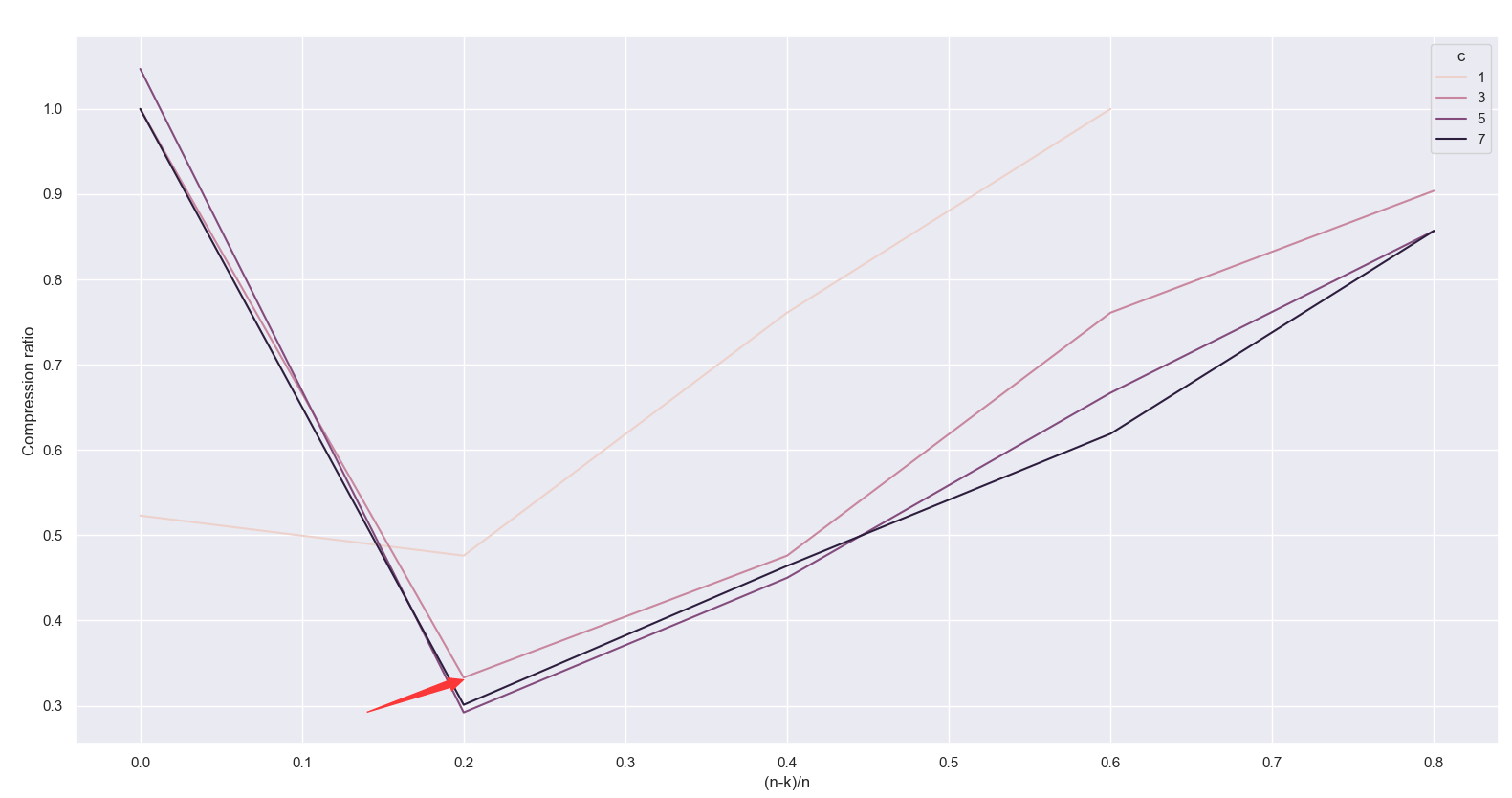


图4-4 启发式算法在T-DK数据集上不同参数的压缩率对比

GD算法相对于其他压缩方法的显著优势在于它提供了对压缩数据的低代价访问能力，相比于其他算法需要全部解压数据后才能进行访问，GD算法通过恢复参数、查询字典对原数据中任意位、任意块进行访问。我们测试了在不同压缩率下访问任意位的代价，该代价以访问的比特位数衡量，结果见图4-5。最坏的情况下（压缩器状态没有恢复），访问单个位需要以访问压缩数据中几百个比特位为代价，但这些代价大多来自于压缩器的状态（参数、字典）；如果压缩器状态已恢复，则访问deviation只需要个位数的访问代价，访问base只需要两位数的访问代价。在牺牲少量的压缩性能的前提下，GD算法提供了远超于传统压缩算法的随机访问性能，对于有着强随机访问需求的场景非常适用。

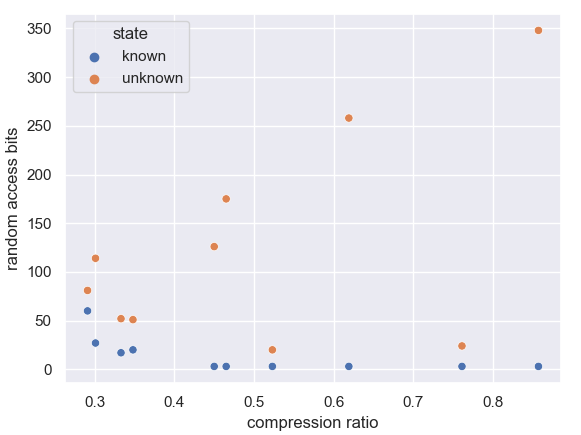


图4-5 不同压缩率下访问单个bit的代价

## 4.4 本章小结

在这一章节中，主要对我们所实现并优化的两个算法Plato和GD进行实验，来测试算法各指标。实验主要安排在二三小节：

第二小节，主要对Plato算法的性能进行测评。首先通过在HA数据集上执行均值、标准差、相关系数等查询，测试了Plato算法给出的误差界与真实误差界的大小，结果表明均值、标准差查询几乎不存在误差，相关系数、自相关的误差较大，但与真实误差在同一量级；然后通过在HB数据集上固定查询为相关系数，测试了不同压缩率下误差界与真实误差界的大小，结果表明误差界虽然比真实误差高1-2个量级，但二者相对于结果都非常小（在水平）。最后比较了使用正交投影性质对误差界做进一步改进前后的误差界变化，结果表明在两段时序数据均值相近时提升较小，但对于两段时序数据均值相差较大的一般情况，误差界有很大改进。

第三小节，主要对GD算法的性能进行测评。首先通过对比GD算法与传统的数据压缩算法GZIP和7z的压缩性能，得出GD算法压缩率略低于GZIP，略高于7z的结论；然后通过手动调整参数并计算压缩率，比较了不同参数下压缩率的大小，结果表明GD算法预处理阶段的参数选择较为合理，比较接近全局最优；最后对GD算法的随机访问效率进行了测试，结果表明压缩器状态未恢复时随机访问需要几百个比特位的代价；但压缩器状态一旦恢复，访问deviation只需要个位数比特的访问代价、访问base只需要两位数比特的访问代价。

# 结 论

在本次毕业设计论文中，我们在边缘硬件上实现了基于有损压缩思想，可对压缩时序数据进行error-bound统计查询的Plato算法和基于无损压缩思想，可对压缩时序数据进行低代价随机访问的GD算法。其中：

1. 对于Plato算法，我们利用了作者提出的误差向量正交投影性质对时序数据乘法并聚集的误差界做了进一步改进。
2. 对于GD算法，通过改进预处理时的算法终止条件避免了预估空间落入局部最优值。

最后通过实验证明了Plato算法给出的误差界具有合理性，且对误差界的改进使均值相差较大的时序数据相关系数查询操作的误差界有很大程度的降低；GD算法的压缩效果比现存的优秀数据压缩算法略差，但牺牲少量压缩率换取的随机访问效果非常可观，且在压缩器状态恢复后的随机访问代价进一步降低。

我们还提出了一些对时序数据压缩查询未来的展望：

1. 随着边缘硬件在智慧城市中的大规模部署，对数据的压缩和查询需求不再拘泥于单机处理，可对分布式情况的压缩与查询做进一步探索。
2. 对于有损压缩，支持近似查询的算法，可以尝试探索拟合函数的更好的性质，对误差界做进一步改进。
3. 对于无损压缩，支持随机访问的算法，可以尝试改进原有启发式算法，使其得出的参数更加合理。

# 参考文献

[1] E. Zivot and J. Wang, “Vector autoregressive models for multivariate time series,” Modeling Financial Time Series S-PLUS, pp. 385–429, 2006.

[2] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock price prediction using the ARIMA model,” in Proc. 16th IEEE Int. Conf. Comput. Model. Simulation 2014, pp. 106–112.

[3] Gourieroux, C., Jasiak, J. and Fol, G. L. (1999). Intra-day market activity. *Journal of Financial Markets* **2**, 193–226.

[4] Bouchaud, J. P., Mezard, M. and Potters, M. (2002). Statistical properties of stock order books: Empirical results and models. *Quantitative Finance* **2**, 251–256.

[5] Smith, E., Farmer, J. D., Gillemot, L. and Krishnamurthy, S. (2003). Statistical theory of the continuous double auction. *Quantitative Finance* **3**, 481–514.

[6] Cont, R. and de Larrard, A. (2011). Price dynamics in a Markovian limit order market. *SSRN eLibrary*.

[7] Cont, R., Stoikov, S. and Talreja, R. (2010). A stochastic model for order book dynamics. *Operations Research* **58**, 549–563.

[8] W. Huang, Y. Nakamori, and S.-Y. Wang, ‘‘Forecasting stock market movement direction with support vector machine,’’ Comput. Oper. Res., vol. 32, no. 10, pp. 2513–2522, Oct. 2005.

[9] K. J. Kim, ‘‘Financial time series forecasting using support vector machines,’’ Neurocomputing, vol. 55, nos. 1–2, pp. 307–319, Sep. 2003.

[10] L. J. Cao and F. E. H. Tay, ‘‘Support vector machine with adaptive parameters in financial time series forecasting,’’ IEEE Trans. Neural Netw., vol. 14, no. 6, pp. 1506–1518, Nov. 2003.

[11] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data,” IEEE Trans. Emerg. Topics Comput. Intell., to be published.

[12] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Benchmark dataset for mid-price prediction of limit order book data with machine learning methods,” J. Forecasting, vol. 37, no. 8, 852–866, 2018.

[13] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Forecasting stock prices from the limit order book using convolutional neural networks,” in Proc. IEEE 19th Conf. Business Inform., 2017, vol. 1, pp. 7–12.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[15] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Trans. Neural Netw., vol. 5, no. 2, pp. 157–166, 1994.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. Advances Neural Inf. Process. Syst., 2014, pp. 3104–3112.

[17] J. Sirignano and R. Cont, “Universal features of price formation in financial markets: perspectives from deep learning,” arXiv preprint arXiv:1803.06917.

[18] C. Carrie, “The new electronic trading regime of dark books, mashups and algorithmic trading,” Trading, vol. 2006, no. 1, pp. 14–20, 2006.

# 哈尔滨工业大学本科毕业设计（论文）原创性声明

本人郑重声明：在哈尔滨工业大学攻读学士学位期间，所提交的毕业设计（论文）《时间序列大数据压缩与查询》，是本人在导师指导下独立进行研究工作所取得的成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明，其它未注明部分不包含他人已发表或撰写过的研究成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。

本人愿为此声明承担法律责任。

作者签名： 日期： 2020 年 6 月 4 日



# 致 谢

冬来春往，寒暑交替。赴哈尔滨求学的第四年，随着毕业论文的行文至此，大学生活也将画上句号。

感谢父母对我从小的辛勤培养，为了让我接受良好的教育不断奔波，我能进入哈工大并顺利完成学业离不开你们的支持。

感谢我的老师高宏教授，在我刚刚笨拙地进入科研的大门时，是您耐心地做我的领路人，从选题到定稿的每一个阶段都少不了您的悉心指导。

感谢原声吉他社的小伙伴们，在我为论文进展缓慢而烦闷时，是你们陪我放松心情，整理好心态重新面对挑战。

最后，感谢所有成长路上给予过我帮助与温暖的亲人、朋友们，我将不负你们的期望，心怀感恩继续前行！