学号\_2014302580124 \_\_

密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

武汉大学本科毕业论文

时间感知的Web服务

质量预测方法研究

院（系）名 称：计算机学院

专 业 名 称 ：软件工程

学 生 姓 名 ：陈秀清

指 导 教 师 ：李兵 教授

二○一八年五月

**BACHELOR’S DEGREE THESIS**

**OF WUHAN UNIVERSITY**

Investigation of time-aware Web service quality prediction method

School (Department) : Computer School

Major : Software Engineering

Candidate : ChenXiuqing

Supervisor : Prof. Bing Li

May 2018

**郑 重 声 明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 日期：

摘 要

Web服务是松散耦合的软件系统，支持跨网络的机器间相互操作交互。服务质量（quality of service,QoS）用来描述Web服务的非功能特性，包括响应时间、吞吐量、代价等。随着服务计算和云计算技术的发展，公开的Web服务的数量每年都在不断增长，而这其中有大量的服务提供相似的功能。面对这么多功能相同或相似的服务，Web服务质量（QoS）已经成为最佳Web服务选取的决定性因素。服务提供者会声明QoS数据并作为服务描述的一部分公开，我们可以由此获得一些静态QoS数据如服务价格等，或者通过服务调用者共享获得一些动态QoS数据如响应时间、吞吐量等。动态的QoS数据在服务运行时具有与时间相关的特性，并非一个平稳的状态，因此，根据过去的观察值或服务调用者共享的QoS数据进行预测，从而满足Web服务推荐系统和面向服务的体系结构中动态服务组合对QoS数据的需求，是服务计算领域需要解决的关键问题之一。

本文针对QoS数据预测开展时间感知的Web服务质量预测方法研究，将时间序列预测技术应用于QoS预测问题上。本文基于一个十组实际服务的QoS数据集构建了一个QoS时间序列数据集，并进行了预处理。在构建完整有效的时间序列数据集的基础上，针对基于ARIMA模型的QoS预测方法和基于LSTM的QoS预测方法展开研究，实现了ARIMA-QoS时序预测模型和LSTM-QoS时序预测模型。为了验证本文实现的模型方法在QoS预测任务上的表现，通过对比实验，分析对比简单移动平均方法、加权移动平均方法和ARIMA-QoS时序预测模型、LSTM-QoS时序预测模型四种不同方法在十组QoS时间序列上的预测结果，综合分析，LSTM-QoS预测模型优于其他三种方法，在QoS预测任务上具有更高的预测精度。

关键词：QoS预测；时序预测；长短时记忆网络；ARIMA模型；

**ABSTRACT**

Web services are loosely-coupled software systems designed to support interoperable machine-to-machine interaction over a network. Quality-of-Service (QoS) is usually employed for describing nonfunctional characteristics of Web services, including response time, throughpout, cost and so on. With the development of Services Computing and Cloud Computing, the number of Web service in growing every year, and there are a large number of services that provide similar functionality. In the face of so many services with the same or similar functions, the quality of service (QoS) has become a decisive factor in the selection of the best Web services. We can get some static QoS values which declared by the service provider and exposed as part of the service description, such as prices, or obtain some dynamic QoS data that shared by service callers, such as response time, throughput, and so on.The dynamic QoS data has time-dependent characteristic in service operation, and is not steady state. Therefore, performing a prediction based on past observed values or the ones shared by service callers to meet the requirements of the QoS data in the Web service recommendation system and the service oriented architecture, is one of the key issue that need to be solved in the field of service computing.

In this paper, time-aware Web service quality prediction method is studied, and time series prediction technology is applied to QoS prediction. This paper constructes a QoS time series based on QoS data obtained by monitoring the execution of 10 real services for 4 months, and then preprocesses the series. After buliding a complete and effective time series, we do a research on QoS prediction based on LSTM model and ARIMA model, finally the ARIMA-QoS timing model model and LSTM-QoS timing prediction model are implemented for the QoS prediction. In order to verify that the model realized in this paper is performed on QoS prediction tasks, the prediction results of four different approaches in QoS time series of 10 real services are analyzed by comparison experiment, comparison of the simple moving average method, the weighted moving average method and the ARIMA-QoS timing prediction model, the LSTM-QoS timing prediction model. Overall, the LSTM-QoS timing prediction model is better than other three methods, and has higher prediction accuracy on the QoS prediction task.

**Key words:**  Q oS prediction; time series prediction; ARIMA model; Long Short Term Memory Network (LSTM)

目 录

[1 绪论 1](#_Toc512971873)

[1.1 课题的研究背景与研究意义 1](#_Toc512971874)

[1.1.1 研究背景 1](#_Toc512971875)

[1.1.2 研究意义 2](#_Toc512971876)

[1.2 国内外研究现状 3](#_Toc512971877)

[1.3 课题的研究内容 4](#_Toc512971878)

[1.4 本文的组织结构 5](#_Toc512971879)

[2 时间序列分析基础 7](#_Toc512971880)

[2.1 时间序列定义 7](#_Toc512971881)

[2.2 时间序列的分解 8](#_Toc512971882)

[2.3 时间序列的平稳性 8](#_Toc512971883)

[2.4 平稳性检验 9](#_Toc512971884)

[2.4.1 特征统计量 9](#_Toc512971885)

[2.4.2 ADF单位根检验 11](#_Toc512971886)

[2.5 本章小结 12](#_Toc512971887)

[3 QoS时间序列构建 13](#_Toc512971888)

[3.1 引言 13](#_Toc512971889)

[3.2 QoS数据时间序列 13](#_Toc512971890)

[3.3 QoS时序数据处理 15](#_Toc512971891)

[3.4 本章小结 16](#_Toc512971892)

[4 基于ARIMA模型的QoS时序预测方法研究 18](#_Toc512971893)

[4.1 ARIMA模型基本结构 18](#_Toc512971894)

[4.2 ARIMA模型预测步骤 19](#_Toc512971895)

[4.3 ARIMA-QoS预测模型 20](#_Toc512971896)

[4.3.1 构建平稳的QoS时间序列 20](#_Toc512971897)

[4.3.2 识别ARIMA模型 21](#_Toc512971898)

[4.3.3 模型拟合评优 22](#_Toc512971899)

[4.4 本章小结 24](#_Toc512971900)

[5 基于LSTM的QoS时序预测方法研究 25](#_Toc512971901)

[5.1 机器学习理论基础 25](#_Toc512971902)

[5.1.1 神经元模型 25](#_Toc512971903)

[5.1.2 目标函数与梯度下降法 26](#_Toc512971904)

[5.1.3 误差反向传播 26](#_Toc512971905)

[5.2 循环神经网络模型结构 27](#_Toc512971906)

[5.2.1 循环神经网络 27](#_Toc512971907)

[5.2.2 长短时记忆网络 28](#_Toc512971908)

[5.3 基于Keras框架实现的LSTM-QoS时序预测模型 29](#_Toc512971909)

[5.3.1 数据预处理 29](#_Toc512971910)

[5.3.2 利用Keras构建神经网络 30](#_Toc512971911)

[5.3.3 网络层结构设计 31](#_Toc512971912)

[5.3.4 模型训练及调优模型 32](#_Toc512971913)

[5.4 本章小结 35](#_Toc512971914)

[6 实验结果分析与对比 36](#_Toc512971915)

[6.1 实验设置 36](#_Toc512971916)

[6.2 结果对比 37](#_Toc512971917)

[6.3 本章小结 40](#_Toc512971918)

[7 总结与展望 41](#_Toc512971919)

[7.1 总结 41](#_Toc512971920)

[7.2 展望 41](#_Toc512971921)

[参考文献 43](#_Toc512971922)

[致 谢 47](#_Toc512971923)

# 1 绪论

## 1.1 课题的研究背景与研究意义

### 1.1.1 研究背景

Web服务是松散耦合的软件系统，通过网络来支持互操作的机器进行交互[1]。随着服务计算和云计算技术的发展，越来越多的企业将自身业务以公开可用的Web服务的方式发布[2]。Web服务的数量每年都在不断增长，而这其中有大量的服务提供相似的功能，比如，据ProgrammableWeb.com统计，能够提供搜索服务的Web API截止2018年4月15日就多达3124个。面对这么多功能相同或相似的服务，用户在选择符合自己的需求的web服务时，就需要考虑web服务除了功能之外的属性，即服务的非功能特性。服务质量（quality of service,QoS）用来描述web服务的非功能特性，包括响应时间、吞吐量、代价等。现实生活中，需要用到某类功能的web服务的用户不可能调用过每个候选服务，而且QoS数据并非是一个平稳的分布状态，服务提供者声明的QoS值有时并不能准确的描述用户即将调用的服务的质量，这就需要服务推荐系统能够通过过去已知的QoS数据预测出未来缺失的QoS数据，从而给用户推荐最快、最便宜的服务。面向服务的体系结构（Service Oriented Architecture,SOA）[3]是一种通过服务之间定义好的接口将不同服务联接起来的组件模型。延迟绑定和发现新服务是SOA的特性，它允许绑定来自服务组合的请求，在这种情况下，选择要调用的服务可能取决于服务的服务质量QoS属性，QoS感知绑定意味着，在功能相同的服务中，人们可以选择最便宜的服务、最快的服务，或者在时间和成本之间实现很好的折衷的服务。对于候选服务，通常会考虑一个可能的QoS值，该值由服务提供者声明，并作为服务描述的一部分公开，或者通过根据过去的观察值进行预测，从而实现对QoS感知的组合和绑定。

时间序列预测技术是一种通过对过去的历史时间序列数据进行分析来实现对未来的一定时期内数值或趋势进行预测的回归预测方法。时序预测理论，从诞生至今，一直在不断得到完善：从最开始提出的自回归（autoregressive,AR）模型，移动平均（moving average,MA）模型到自回归滑动平均（Auto Regressive and Moving Average,ARMA）模型，这些模型是时间序列分析研究的奠基石。而后Box和Jenkings在回归分析法的基础上，系统地阐述了求和自回归移动平均（autoregressive integrated moving average,ARIMA）模型的识别、评估、检验及预测的原理和方法[4]，奠定了传统时间序列分析的里程碑。随着时序预测模型研究的深入，传统时序预测模型的识别、定阶、参数的估计、模型检验等机制都逐渐得到了完善，但是由于模型理论相对简单，预测只是简单的认为未来数据保持了过去趋势，面对复杂的应用环境时，模型表现不佳预测精度不够高，传统时序预测的发展遇到了瓶颈。而随着机器学习和大数据的兴起与发展，基于机器学习的算法开始被应用于时序预测模型中。神经网络是由具有非线性作用函数（即激活函数）的神经元组成的广泛并行互连的网络，具有高度的运算能力和容错能力。根据实际的应用场景，将重要特征引入模型，设定神经元的参数权重，并在训练过程中调整，从而重建时间序列中各个元素间的联系，建立输出模式和输出模式之间的关联，实现历史时序到未来时序的映射，从而达到预测未来的目标[5]。

### 1.1.2 研究意义

随着Web服务不断增加，很多时候QoS决定了用户选取多个功能相同的web服务中的哪一个。目前研究者们在Web服务推荐[6][7]、服务动态组合[8]、面向服务的体系结构构建等方面提出了很多方法，而这些方法都需要准确的QoS数据提供重要支撑。服务提供者会声明QoS数据，并作为服务描述的一部分公开，我们可以由此获得一些静态QoS数据如服务价格等，或者通过服务调用者共享获得一些动态QoS数据如响应时间、吞吐量等。动态的QoS数据在服务运行时具有与时间相关的特性，并非一个平稳的分布状态，因此，根据过去的观察值或服务调用者共享的QoS数据进行预测，从而满足web服务推荐和面向服务的体系结构中动态服务组合对QoS数据的需求，是服务计算领域需要解决的关键问题之一，而目前在QoS预测问题中应用较广泛的协同过滤技术研究局限于根据时间片段内的历史QoS数据进行预测，未考虑时间维度下的QoS数据的动态变化，往往导致预测值与实际值发生较大误差[9]。

时序预测是数理统计和大数据分析领域的重要研究之一，时序预测是根据现有的历史数据，发掘其中隐含的规律，从而预测未来。传统时序预测具有一整套完善的理论、方法，并且被广泛应用于各个领域，随着大数据和机器学习的兴起和发展，使用神经网络对时间序列进行建模分析开始得到了关注，构建一个神经网络模型，投入训练的数据量越大，模型就能学习到更多的信息，得到的时序预测模型准确度就越高。目前，时序预测已经广泛应用于科学领域和工业领域中，有效的时序预测具有重大的研究意义和实用价值，比如，在金融市场中，产品的价格变化可以建模时间序列，然后通过时序预测模型预测得到序列未来的值，那么投资者就可以从中获得最大收益，此外，时序预测也在工业生产，交通管理等领域中的得到了广泛的应用。提高时间序列预测模型的性能和效率，对其应用领域的管理控制和重大决策具有极大的意义。

综上所述，QoS预测任务是一项极具挑战性的问题，而准确有效的QoS预测为web服务推荐和SOA面向服务的体系结构的服务组合提供了重要支撑，而将时序预测技术应用于QoS预测问题上，有望提高QoS预测精度，实现利用历史QoS数据预测未来QoS数据，这是富有挑战性同时也具有研究意义的工作。

## 1.2 国内外研究现状

1927年， G.U.Yul为了预测市场变化规律提出自回归模型（autoregressive，AR），1931年， G.T.Walker将移动平均模型（moving average, MA）和自回归移动平均模型（autoregressive moving average, ARMA）应用于分析印度大气规律，这些模型奠定了时间序列分析方法的基础。1970年，G.E.P.Box和G.M.Jenkins提出了求和自回归移动平均（autoregressive integrated moving average，ARIMA）模型以及一整套的建模，估计、检验和控制方法[4]。Engle在1982年提出了自回归条件异方差（ARCH）模型用以研究英国通货膨胀率的建模问题[10]。由于基础的传统时序模型在预测机制上存在缺陷，模型简单默认未来保持了过去的趋势，直接套用模型，得到未来的数据，模型的泛化能力差，面对复杂的应用环境容易出现预测不准确的情况。因此很多时序研究者采取时序分析与某种算法结合的机制，对时序数据混合建模，采用卡尔曼滤波算法结合时间序列分析法，基于传统时序预测模型，结合次优近似估计算法，通过更新用于建模的数据，实现了动态递推建模，方法简单，易于实现[11][12]。

基于机器学习的时序预测模型，通常利用神经网络或者支持向量机等方法，通过学习大量的时序数据中的信息，建立预测模型并评估，实现对未来数据的预测。Chen等利用回归神经网络进行预测和交易[13],Kwon等提出了一种混合神经网络来预测股票[14]，Chao等展示了利用SVM对金融时间序列预测的结果，该结果显著优于BP神经网络的结果[15]。JianZhou提出基于小波消噪的BP神经网络预测模型，利用小波分解技术提高了混沌时序数据的神经网络模型的预测精确度[16]。Naeini等通过上证综指的预测实验证明了多层感知神经网络模型在股票预测中具有巨大的潜力[17]。Yoo等通过实验比较不同的时序预测模型[18]，证明了支持向量机能够很好地识别时序的趋势以及回归分析，而神经网络在表达线性关系方面精确性更高。

目前，协同过滤方法在服务质量(QoS)预测中应用很广泛：基于用户的协同过滤方法预测QoS根据与目标用户相似的邻居用户的评价记录的分析，预测出目标用户调用web服务时的QoS[19]。基于用户和基于项目的协同过滤方法提出一种混合协同过滤算法，通过分配不同的权重系统地结合了基于用户和基于项目的方法，提高了预测的精确度[20]。然而，基于协同过滤的QoS预测研究局限于根据某一段时间内服务的历史QoS数据进行预测，没有考虑到QoS值在时间维度上的变化，仅将当前时间段内的QoS值计算结果作为预测结果，无法对未来一定时期内的QoS值进行预测，而且这些方法不能反映web服务随时间变化的趋势对未来的QoS值的影响，从而导致了预测的结果和真实值存在较大的误差。为此，Chen Lei-lei等提出了服务的QoS周期模型,分析QoS随时间的变化规律，并利用该模型实现QoS预测[21]。Hua Zhe-bang等使用ARMA模型和ARIMA模型在一段时间内对QoS数据进行时序拟合分析[22]，实现了未来一定时间内QoS值的有效预测，但是该方法在预测时需要对QoS逐一建模，而且在识别模型参数时采用了穷举的方式使得算法复杂度随QoS数量呈指数增长。

## 1.3 课题的研究内容

本文的课题围绕时间感知的Web服务质量预测方法展开研究，针对QoS数据预测进行了研究。本文将注意力集中在响应时间上，该数据集来源于Bice Cavallo的实证研究[23]，在长达4个月的时间里每隔一个小时调用10个实际服务收集到的监测QoS数据，然后记录（1）服务是否可用，（2）SOAP输出信息（3）观察响应时间（4）吞吐量，作为SOAP消息的大小除以响应时间（5）服务是否异常。本课题主要针对ARIMA模型方法，机器学习（LSTM网络）模型方法进行理论分析与实证研究，此外，为了对比实验，实现了简单移动平均方法、加权移动平均方法，进行评测对比实验验证各模型方法在QoS预测问题上的表现。本文具体的研究内容包括：

（1）使用QoS数据集的响应时间建模，考察观察值序列的特征，对数据进行预处理，如缺失值处理，异常值处理，为后续的实验研究奠定基础。

（2）基于QoS数据预测问题开展ARIMA模型方法的研究与实现，根据序列的特征选择适当的拟合模型，检验模型，优化模型，评估模型在QoS预测问题上的表现。

（3）基于QoS数据预测问题开展机器学习模型方法的研究与实现。本文针对响应时间序列的特性，以循环神经网络(RNN)及其变种长短时记忆网络（LSTM）理论学习为基础，设计适应该时间序列的LSTM网络模型并实现，利用已有的数据集进行训练，评估，验证该模型方法在QoS预测问题上的表现。

（4）进行对比实验，分析比较不同的QoS预测方法在本数据集上的表现。

## 1.4 本文的组织结构

第一章是绪论，主要介绍了课题的研究背景和研究意义，以及国内外对课题的研究情况，最后概述了针对课题的研究本文拟开展的研究主要方面。

第二章是时间序列分析基础，阐述了时序分析的相关理论基础，对时间序列的分解、时序的平稳性及其检验做了详细的介绍。

第三章是QoS时间序列构建，分析课题研究所用的QoS数据集，针对响应时间构建QoS时序数据集，考察序列的特征，进行数据预处理，为后续的实验提供完整有效的QoS时序数据集。

第四章是基于ARIMA模型的QoS时序预测方法研究，简述了ARIMA模型的理论基础，实现了适应于本课题的QoS预测任务的ARIMA-QoS时序预测模型，并评估模型的预测结果。

第五章是基于LSTM的QoS时序预测方法研究，首先阐述了本文研究所使用的相关神经网络模型：循环神经网络模型RNN和长短时记忆网络模型LSTM的原理和结构，随后基于Keras框架构建了LSTM-QoS时序预测模型，训练网络并调参，利用LSTM-QoS模型预测未来一定时期内的QoS值。

第五章是实验结果对比分析。针对十组QoS数据，分析对比简单移动平均、加权移动平均方法、基于ARIMA模型的QoS预测模型和LSTM-QoS时序预测模型四种不同预测方法的预测结果。

第六章是总结与展望，对本文的研究进行了总结并概述后续工作的发展方向。

# 2 时间序列分析基础

## 2.1 时间序列定义

统计学定义，按时间顺序排列的随机变量序列[24]

， ，…，，… （2.1）

表示时间序列，用

，，…，，… （2.2）

表示上述序列的n个有序观察值，则称（1.2）为（1.1）的一次实现或一条轨道。

在现实生活中，通常我们得到的数据是时间序列的有限观测样本（1.2）.即在实际问题中，我们是对某一个（组）变量进行观察测量，在一系列时刻,,…所得到的离散的数值组成的集合，称为时间序列。 比如某只股票从2017年1月1日到2018年1月1日之间各个交易日的收盘价，可以构成一个时间序列；把1999-2017年湖北省高考报考人数按年份记录下来，可以构成一个时间序列，等等。日常生活中，时间序列数据比比皆是，无论在科学研究，工业生产还是经济金融、医疗卫生等等领域中都有时间序列的存在与应用。时序分析的主要任务就是考察观察值序列的特征，选择适当的拟合模型，然后利用模型的统计特性去揭示数据的统计规律，以期望达到控制或预报的目的。

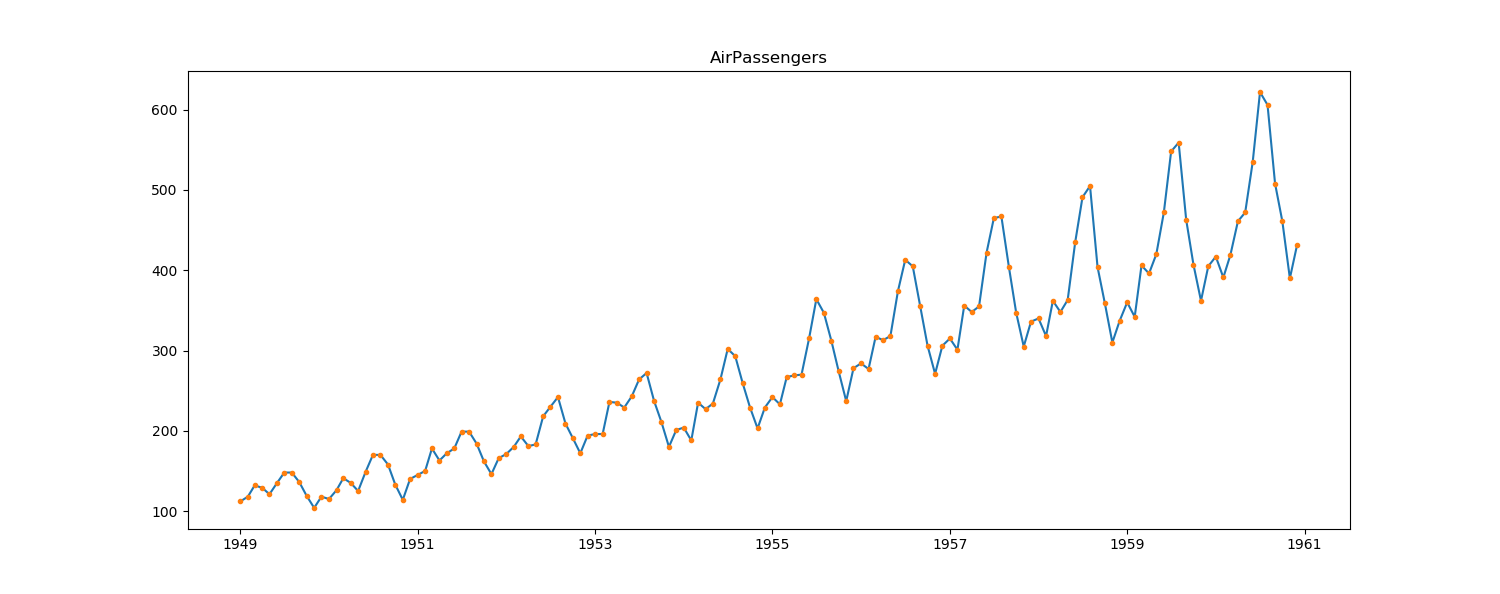


图2.1 某航空公司1949年—1961年每月乘客数据

如图2.1为某航空公司1949年到1961年每月的乘客数据，是一条典型的时间序列数据，其中横（X）轴代表时间，纵（Y）轴代表每月乘客数量，航空公司可以根据该图结合相关知识与实际物理资源分配，考虑增开或者缩减航班以获得最大的收益。

## 2.2 时间序列的分解

任何时间序列经过合理的函数变换后都可以认为是由趋势项、周期项、随机噪声项这三个部分组成的[25]。趋势项就是时序在一定长的时期内呈现出持续增或持续降的波动，周期项是时间序列由于受到某些外在因素如气候变化、节假日风俗等影响造成的在一定时期内重复出现的周期性波动，或呈现出非固定长度的涨落相同的交替波动。时间序列滤去趋势、周期波动后仍然有随机噪声，也就是呈现随机波动的平稳序列。我们经过适当的函数变换，把时间序列分解成三个部分的叠加：

= + + ，t=1, 2, … （2.3）

其中是趋势项，是周期项，是随机项，时间序列是三者的叠加。

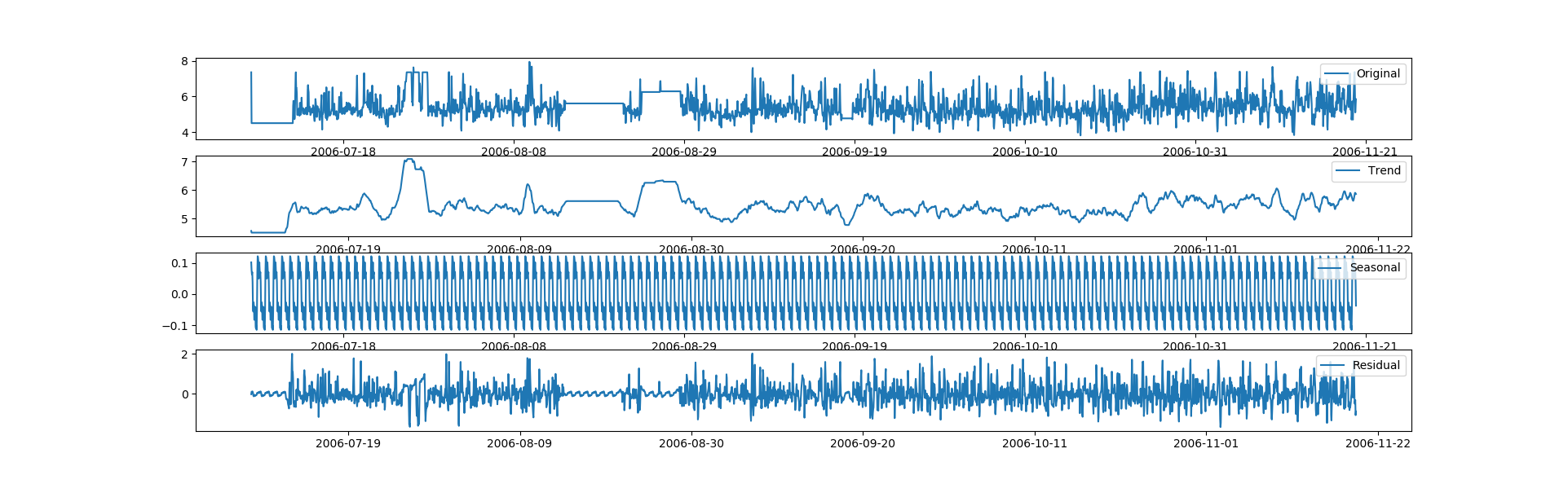


图2.2 QoS时间序列的分解

如图2.2可以明显的看到，我们将原始数据（Original）分解成了三部分：趋势项（Trend）数据，针对QoS响应时间构成的序列，没有太明显的递增或递减的趋势；周期项（Seasonality）数据,具有明显的由低到高又由高到低的涨落交替的周期性；随机项（Residual）数据，去除了长期趋势和周期性变化之后的由于其他各种因素的综合影响，呈现出随机波动的平稳序列。

## 2.3 时间序列的平稳性

如上节所述，时间序列可以拆分成趋势项、季节项和随机项，而随机项往往表现出某种平稳波动性，称之为平稳序列。因为平稳时序的历史， ，…，…中往往含有的信息，所以可以使用平稳时间序列的历史数据来预测未来的数据。根据限制条件的严格程度，平稳时间序列分为下面两种[24]：

（1）严平稳序列

设{}为一时间序列，如果对任意正整数n和, 随机变量 和同分布，则称该序列为严平稳序列。

（2）宽平稳序列

宽平稳序列利用特征统计量作为判断标准：若时间序列{}的均值关于时间是一个常量，自协方差函数和自相关系数只依赖于时间的平移长度而不与时间的起止时间点相关[26]，即对于任意时刻t, 任意整数i,与的协方差不随时间而改变，则称其为宽平稳序列。

实际生活应用中，随机序列的联合分布往往难以获得，另一方面，已知其联合分布的情况下，计算和应用中仍存在很多困难。所以严平稳序列只具有理论意义，而实际应用中一般使用宽平稳序列。

## 2.4 平稳性检验

### 2.4.1 特征统计量

宽平稳认为序列的统计性质主要由它的低阶矩决定，特别是均值、方差、自协方差和自相关系数，它们也统称为特征统计量[24]。由于它们具有明显的统计意义，能代表序列的主要统计特征，而且计算简便，所以我们可以利用特征统计量识别随机序列的性质。判断时间数据是稳定的一个简单方法是基于对于时间是常量的几个统计量：均值常量，方差常量，自协方差与时间独立。

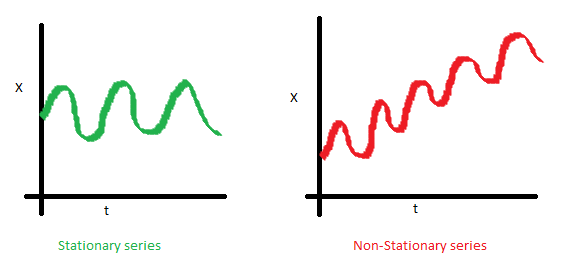
（1）均值：

对时间序列{}，在任意时刻t, 序列在t时刻的均值函数定义为：

= （2.4）

其中 为 的分布函数且<，均值函数序列{}反映的是时间序列{}每时每刻的平均水平。

如图2.3所示，X是时间数据，t是时间，图2.3(a)中时间数据的均值对时间而言是一个常量，所以X是稳定的；图2.3(b)中时间数据的均值很明显具有时间依赖，均值是时间的函数，随着时间变化，数据呈现上升的趋势，是非稳定的。



(a) (b)

图 2.3 平稳时间序列和非平稳时间序列的均值对比

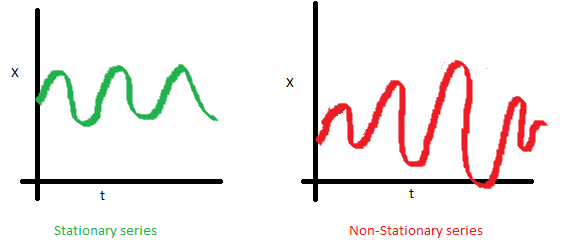
（2）方差：

当< 时，定义时间序列的方差函数为：

D = E= （2.5）

其中 为 的分布函数且<}是的均值函数，方差函数序列反映的是时序围绕其均值做随机波动的平均波动程度。

如图2.4（a）所示，时间数据的方差对于时间是常量，即数据的值域围绕着均值上下波动的振幅是固定的，所以数据是稳定的；而图2.4(b)中，时间数据的振幅在不同时间点不同，所以方差对时间不是独立的，数据是非稳定的。



(a) (b)

图 2.4 平稳时间序列与非平稳时间序列的方差对比

（3）自协方差函数和自相关系数：

对于时间序列{, t},任取t, s T, 定义(t, s) 为序列的自协方差函数：

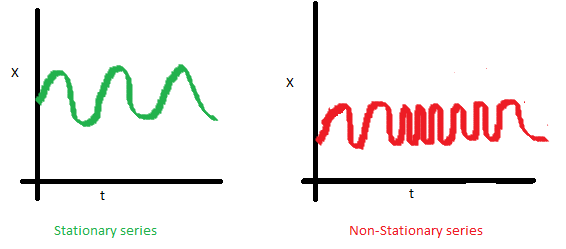
(t, s) = E（）（） （2.6）

定义(t, s)为序列的自相关系数，记为ACF：

(t， s) = （2.7）

自协方差函数和自相关系数度量的是同一事件在不同时期之间的相关程度，也就是度量自己过去的行为对自己现在的影响[27]。

如图2.5（a），时间数据在两个不同时刻的自协方差与时间无关，所以它是稳定的。图2.5（b）中，随着时间的不同，数据的波动频率明显不同，导致两个不同的时刻 ,每次的取值不同时，就会得到不同的协方差，所以是非稳定的。



(a) (b)

图 2.5 平稳时间序列和非平稳时间序列的自协方差对比

### 2.4.2 ADF单位根检验

单位根检验就是检验序列中是否存在单位根，若存在则为非平稳序列[28]。对于时间序列，可以用如下自回归模型检验单位根：

= + （2.8）

其中为误差，零假设和备择假设分别为：=1，（非平稳），：<1，（平稳）。在零假设成立条件下，用DF统计量进行单位根检验，若用样本计算得到的DF>临界值，则接受，非平稳；若DF 临界值，则拒绝，是平稳的[29]。

ADF是一种常用的单位根检验方法，在一定置信度下，原假设为时序存在单位根，序列非平稳。而对于一个平稳的时间序列，就需要在给定的置信水平上显著地拒绝原假设[30]，即通过检验值临界值来拒绝（原假设），认为时间数据是平稳的，否则是非平稳。根据ADF检验的原理，可以从两个方面考察验证结果：

（1）ADF统计量显著小于临界值。如果得到的统计量显著小于3个置信度（1%，5%，10%）的临界值时，说明是拒绝（原假设）的。

（2）p值非常接近于0（4位小数基本即可）。根据统计学经验，p值越小则对应时间序列具备越好的平稳性。

如表2.1所示，以Amazon的QoS数据的ADF检验结果为例，首先，p值为0.000002 ,可以认为非常接近于0；然后比较三个不同程度拒绝（原假设）的临界值与ADF统计量的大小，很明显，1%，5%，10%水平下的临界值均明显大于ADF统计值，所以可以认为它显著的拒绝（原假设），所以Amazon的QoS数据构成的时间序列是平稳的。

表2.1 十组实际服务的QoS数据ADF检验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 服务 | ADF统计量  (Statistic) | 临界值  (1%） | 临界值  （5%） | 临界值  （10%） | p值 |
| Amazon | -5.529368 | -3.432593 | -2.862531 | -2.567298 | 0.000002 |
| BLiquidity | -9.6988e+00 | -3.4314e+00 | -2.8620e+00 | -2.5670e+00 | 1.0854e-16 |
| Currency-  Converter | -6.1548e+00 | -3.4329e+00 | -2.8626e+00 | -2.5673e+00 | 7.4115e-08 |
| FastWeather | -6.0612e+00 | -3.4346e+00 | -2.8634e+00 | -2.5677e+00 | 1.2113e-07 |
| GetJoke | -4.309718 | -3.432958 | -2.862692 | -2.567383 | 0.000427 |
| Google | -5.198455 | -3.432592 | -2.862531 | -2.567297 | 0.000009 |
| Hyperlink-  Extractor | -6.7564e+00 | -3.4324e+00 | -2.8624e+00 | -2.5672e+00 | 2.8642e-09 |
| QuoteOfTheDay | -7.5815e+00 | -3.4325e+00 | -2.8625e+00 | -2.5672e+00 | 2.6731e-11 |
| StockQuotes | -6.2764e+00 | -3.4325e+00 | -2.8625e+00 | -2.5672e+00 | 3.8907e-08 |
| XMLDailyFact | -5.9659e+00 | -3.4329e+00 | -2.8626e+00 | -2.5673e+00 | 1.9873e-07 |

## 2.5 本章小结

本章主要介绍了时间序列、时序分析的理论基础，首先简单介绍了时间序列的定义，然后通过对时间序列的分解对时序的周期性、趋势性和随机性进行了描述，最后重点分析了平稳序列和如何利用特征统计量或ADF单位根检验验证平稳序列。

# 3 QoS时间序列构建

## 3.1 引言

数据是模型方法研究的基础，现实生活中，需要用到某类功能的web服务的用户不可能调用过每个候选服务，而且QoS数据并非是一个平稳的分布状态，服务提供者声明的QoS值有时并不能准确的描述用户即将调用的服务的服务质量，这就需要服务推荐系统能够通过过去已知的QoS数据预测出未来缺失的QoS数据，从而给用户推荐最快、最便宜的服务。本文以服务质量（QoS）作为研究对象，使用Bice Cavallo在文献[23]的实证研究中收集到的QoS数据集：在长达4个月的时间里每隔一个小时调用10个实际服务监测到的QoS数据，然后记录（1）服务是否可用，（2）SOAP输出信息，（3）服务响应时间，（4）吞吐量，SOAP消息的大小除以响应时间的所得结果，（5）服务是否异常。分析观测数据的组织结构和经过提取后的QoS数据的特征，并以此数据集为基础，开展后续的研究实验。

## 3.2 QoS数据时间序列

服务质量QoS用来描述web服务的非功能特性，包括响应时间，吞吐量，代价等。本文针对QoS数据预测进行研究，使用的数据集来源于Bice Cavallo的实证研究[23]，在长达4个月的时间里每隔一个小时调用10个实际服务收集到的观测QoS数据，数据集的数据结构如表3.1所示：

表3.1 十组实际服务的观测QoS数据集数据结构表

|  |  |  |
| --- | --- | --- |
| 列 | 数据项名 | 注解 |
| 0 | time | 调用服务的时间 |
| 1 | MID | 请求的服务器的ID |
| 2 | MRT | 响应时间 |
| 3 | Throughput | 吞吐量，消息大小除以响应时间 |
| 4 | Unable | 请求的服务是否异常 |

从表3.1可知，调用10个实际服务收集到的QoS数据中存在服务异常的情况，因此，我们从csv中获得的原始数据存在一定缺陷，需要对其进行处理才能提交给后续实验进行时序分析与预测。由于存在服务异常的情况，原始数据中缺失部分时间段的数据，对于缺失数据一般的处理方法如下：

（1）滤除缺失数据：最常见也最简单，在数据量比较大而缺失数据所占比例比较低时十分有效，但在样本较少时，会严重破坏数据的客观性，影响结果的准确性，为了确保后续时序预测时的一致性，本文不采取这种方法。

（2）均值替换法：就是用其他有效的样本数据的平均数对缺失的样本进行替换，是一种较为简单快速的方法，不影响样本的均值估计但是会降低其标准差和方差[31]。

（3）最近数据填充：就是在完整数据中找到一个和它最相似的样本，考虑数据的延续性，一般使用离缺失样本最近的一个有效样本数据填充，根据实际问题，可以选择前一个或后一个数据代替缺失值。

（4）插值法填充：插值法就是通过两点（，），（，）来估计中间点的值，使用缺失数据的前一个和后一个有效数据的均值来代替缺失值。

（5）算法拟合填充：利用某种回归模型拟合已有数据，即基于数据集，建立回归方程，用来进行未知特征值的预测，使用预测结果填补原缺失数据。

获取原始数据，进行缺失值处理后，本文将注意力集中在响应时间上，对十组QoS数据集的响应时间建模，考察观察值序列，十组QoS的统计特征如表3.2所示：

表3.2 十组实际服务的QoS数据的统计特征

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 服务 | 均值  (mean) | 标准差  (std） | 最小值  （min） | 最大值  （max） | 中位数  （50%） |
| Amazon | 6.528037 | 4.136224 | 2.363000 | 97.998000 | 5.324000 |
| BLiquidity | 2.472603 | 2.804451 | 1.241000 | 99.939000 | 1.559000 |
| Currency-  Converter | 9.659897 | 24.248706 | 1.931000 | 311.519000 | 4.606000 |
| FastWeather | 3.586219 | 2.789075 | 1.664000 | 25.636000 | 2.383000 |
| GetJoke | 2.648663 | 1.640033 | 1.708000 | 30.800000 | 2.035500 |
| Google | 3.522718 | 5.578086 | 1.670000 | 160.403000 | 2.317000 |
| Hyperlink-  Extractor | 3.772606 | 3.186908 | 2.246000 | 97.959000 | 2.970500 |
| QuoteOfTheDay | 3.816478 | 2.180275 | 1.915000 | 30.378000 | 3.992000 |
| StockQuotes | 2.594468 | 3.156701 | 1.783000 | 101.800000 | 1.936000 |
| XMLDailyFact | 1.997373 | 1.445478 | 1.376000 | 21.616000 | 1.556000 |

## 3.3 QoS时序数据处理

如表3.2所示，经过缺失值处理之后的数据仍然具有一定缺陷，很明显地，均值和最小值、中位数之间的距离比较小，这是正常的，但是最大值与均值、中位数之间的距离很大，特别地，最大值与平均值的偏差已经远大于三倍标准差，在统计学上，“3σ原则”将测定值与平均值的差距大于倍标准差的值称为异常值，由于“3σ原则”只适用于服从正态分布的数据，本文采用另一种异常值检测方法——箱线图，它对数据分布没有特殊要求，箱线图将大于或小于箱线图设定的上下界的观察值视为异常值：

{outlier} = {X>Q3+1.5Q}{X<Q1-1.5Q} （3.1）

其中Q1，Q3分别代表将所有数据按从小到大排列之后的第25%位数（下四分位），第75%位数（上四分位），Q是四分位数间距即Q3和Q1的距离[32]。我们可以更直观地通过时序图和箱线图观察到数据的异常值情况。

我们以服务QuoteOfTheDay为例，图3.1、图3.2分别为该服务经过缺失值处理后的响应时间的时序图和箱线图。我们可以从图3.1中很直观地观察到QoS数据具有波动而且有一些特别突出的波峰，根据实际应用经验，用户在同一网络环境下调用同一web服务，不同时间点的服务响应时间偏差是在一定范围内的，如果某个时刻调用该服务的响应时间比正常超出许多，考虑可能有异常情况。图3.2中标出了Q1，Q3，中位数，以及箱线图的上下边界。图中还存在着很多很刺眼的离群点，这是由于原始数据中存在着特别大的异常值。箱线图使用具有一定耐抗性的四分位数作为判断异常值的标准，检测结果比较客观，所以使用箱线图检测异常值具有一定优越性[33]。检测异常值之后，考虑异常值的处理方法，先将异常值处理为缺失数据，然后根据3.2节中所述的缺失值处理方法进行处理，至此，我们得到了符合后续时序分析实验要求的QoS时序数据。

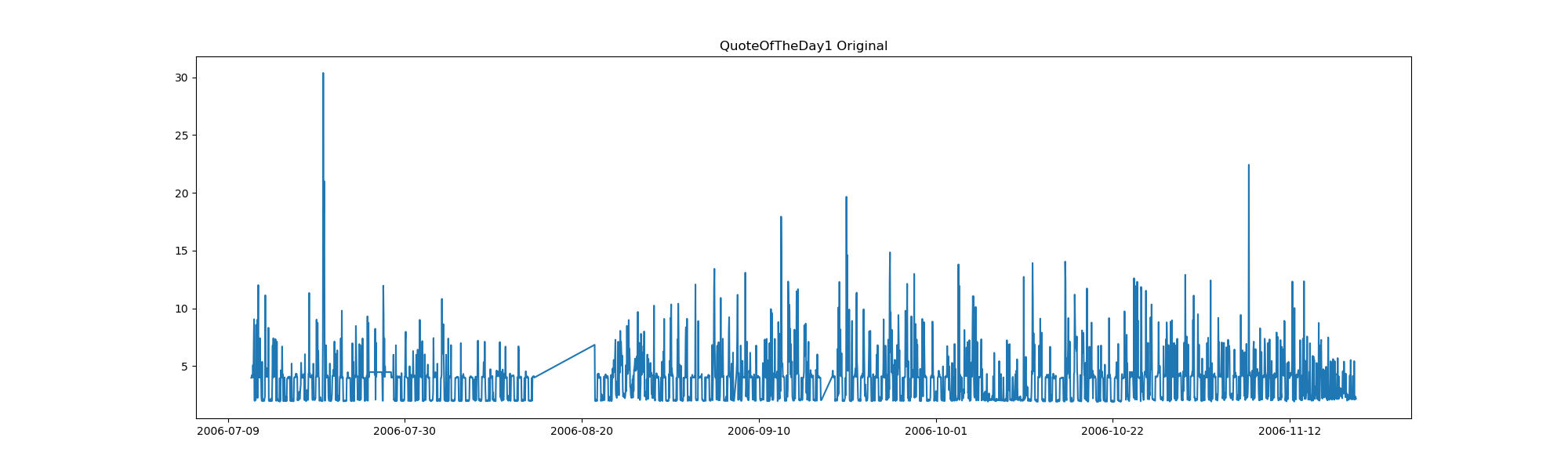


图3.1 QuoteOfTheDay服务的QoS数据时序图

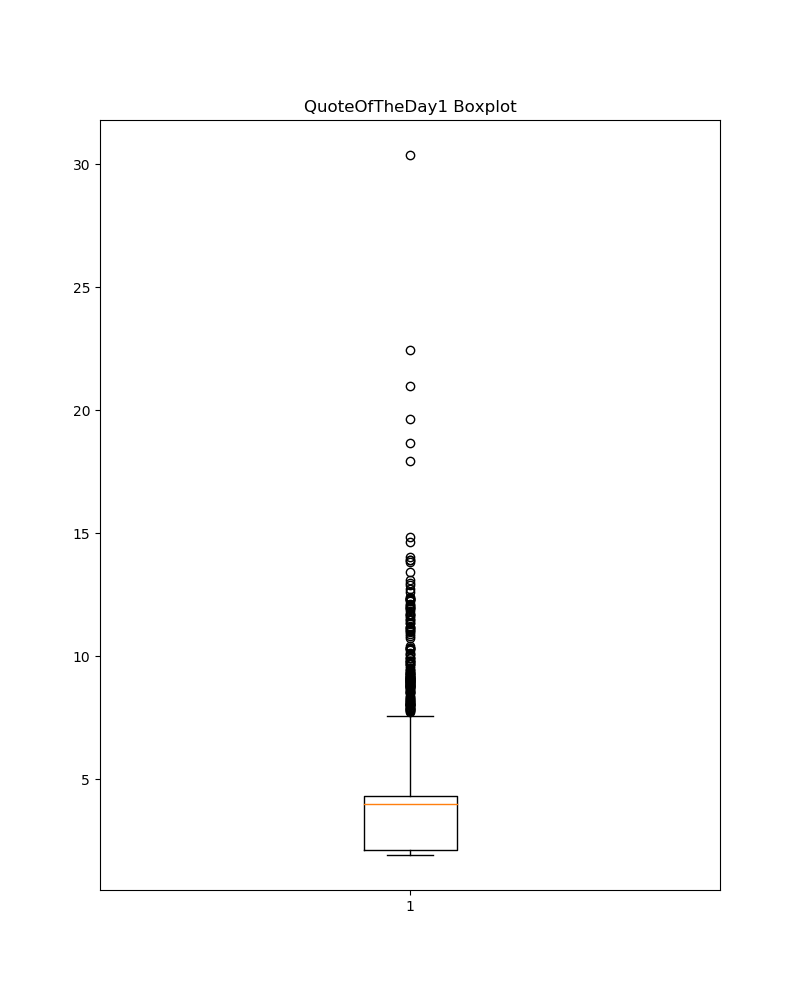


图3.2 QuoteOfTheDay服务的QoS数据箱线图

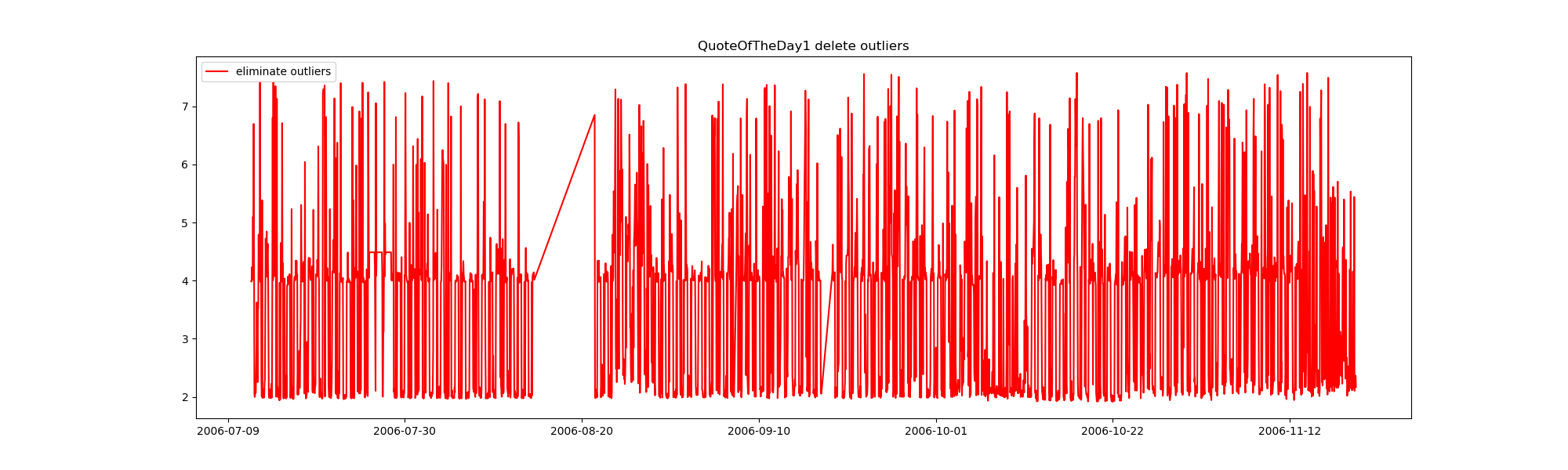


图 3.3 QuoteOfTheDay服务的QoS数据异常值处理后的时序图

图3.3 是QuoteOfTheDay服务的QoS数据经过异常值处理后的时序图，对比图3.1我们可以清楚的看到，经过异常值处理后的QoS时序波动在一个合理的范围内，没有了尖锐的异常值影响时间序列的分布。

经过处理后的QoS时间序列数据严格保证了完整性和有效性，Bice Cavallo在实验中收集的数据以“时”为频率单位进行收集的，监测时间长达四个月，对于每个实际服务而言，具有3000多条数据，我们所构建的时间序列数据集包含3万条QoS响应时间数据记录，为课题后续研究奠定了基础。

## 3.4 本章小结

本章针对本文的研究，使用Bice Cavallo在文献[23]所述的实证研究中收集到的QoS数据集建立了一个十组实际服务的QoS（响应时间）时间序列数据集，通通过对QoS数据的结构和统计特征进行分析，对原始数据进行处理，最终构建了可用于后续时序分析的完整有效的时间序列模型。

# 4 基于ARIMA模型的QoS时序预测方法研究

## 4.1 ARIMA模型基本结构

ARIMA模型全称为自回归求和移动平均模型，是一种经典的时序预测模型，是由Box和Jenkins在70年代提出[34]，其基本思想是将时间数据视为一个随机序列，然后使用适当的数学统计模型描述这个序列，识别模型后就可以使用这个模型通过历史值来预测未来值[35]。ARIMA（p,d,q）称为差分自回归移动平均模型，其中d是为了得到平稳序列的进行差分的阶数，AR（p）是自回归，p为回归项，MA（q）是移动平均，q为移动平均项[25]。

（1）AR模型：

如果{是白噪声，即纯粹的随机过程，具有不随时间变化的恒定均值和方差，当满足以下条件时，过程{}称为p阶自回归过程，简称AR（p）:

=+…++ （4.1）

其中{}是常量，且不为零保证了模型的最高阶数为p，AR过程认为当前值是过去值的加权求和加上一个随机波动的偏差，承认了过去对现在的影响。

（2）MA模型：

设{}是白噪声，具有不随时间变化的恒定均值和方差，当满足以下条件时，过程{}称为q阶移动平均过程，简称MA（p）:

=++…+ （4.2）

其中{}为常量，MA过程认为当前值是过去的白噪声的移动平均。

（3）ARMA模型：

一般来说，MA（q）代表了q个随机脉冲的混合，AR(p)认为过去p个事件构建了现在，将q阶MA过程和p阶AR过程组合成一个即考虑了过去事件又考虑了随机脉冲影响的ARMA（p,q）模型：

=+…++++…+ （4.3）

其中{}是原始时间序列，{}是服从正态分布的随机脉冲。

（4）ARIMA模型：

ARMA过程对处理的数据有平稳性要求，为了处理非平稳时序，ARIMA过程引入了d阶差分，将式（3.3）中的用替代，定义为：

= = （4.4）

其中B定义为：

= （4.5）

则ARIMA（p,d,q）可以定义为：

=+…++++…+ （4.6）

## 4.2 ARIMA模型预测步骤

ARIMA模型预测的基本流程如图4.1所示：

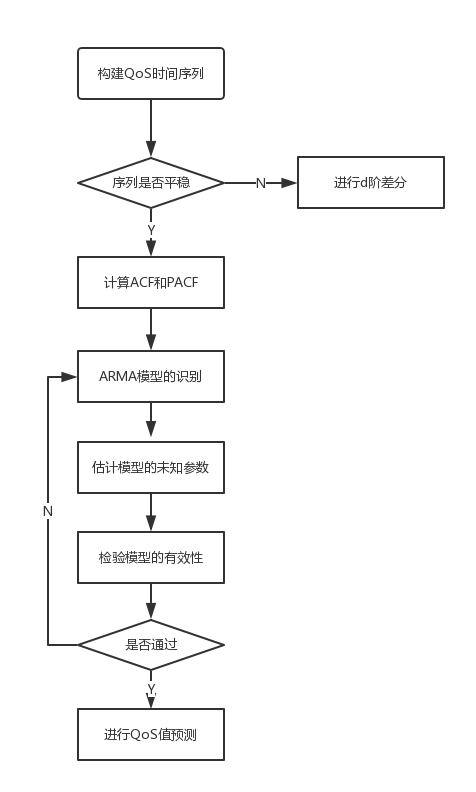


图4.1 ARIMA模型预测基本流程

首先是对序列的平稳性进行识别，根据2.4节，可以根据时序图以及ADF单位根检验判断序列是否平稳。如果序列是非平稳的则需要对序列进行差分处理。然后利用ARIMA模型的识别规则，根据ACF/PACF图进行参数估计，建立相应的模型，进行假设检验，判断残差序列是否为纯随机过程，最后使用检验合格的模型进行QoS值预测。

## 4.3 ARIMA-QoS预测模型

4.3.1 构建平稳的QoS时间序列

在第3章中，我们构建了十组完整有效的QoS时间序列，索引为调用时间，频率是小时，预测目标是响应时间，以Amazon为例，其经过缺失值和异常值处理之后的时序图如下：

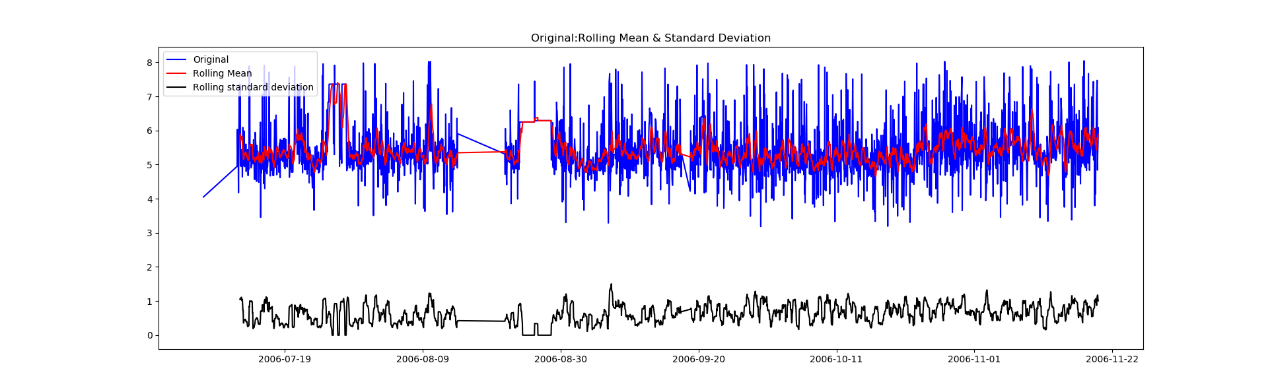


图4.2 Amazon QoS 样本数据及其均值、标准差时序图

Amazon QoS原始时间数据的ADF检验如表2.1所示，t统计量在99%置信度下是显著的，但是其均值序列波动比较大，对其分别进行一阶差分、二阶差分之后的时序图如图4.3、4.4所示：

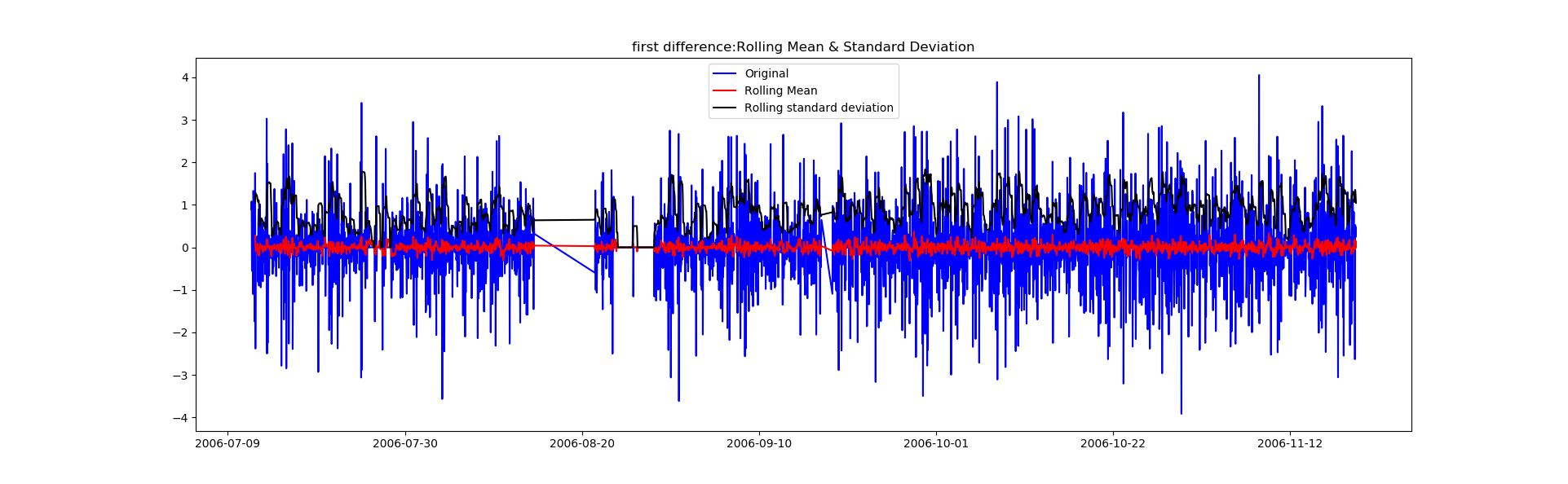


图4.3 Amazon QoS一阶差分序列及其均值序列、标准差序列

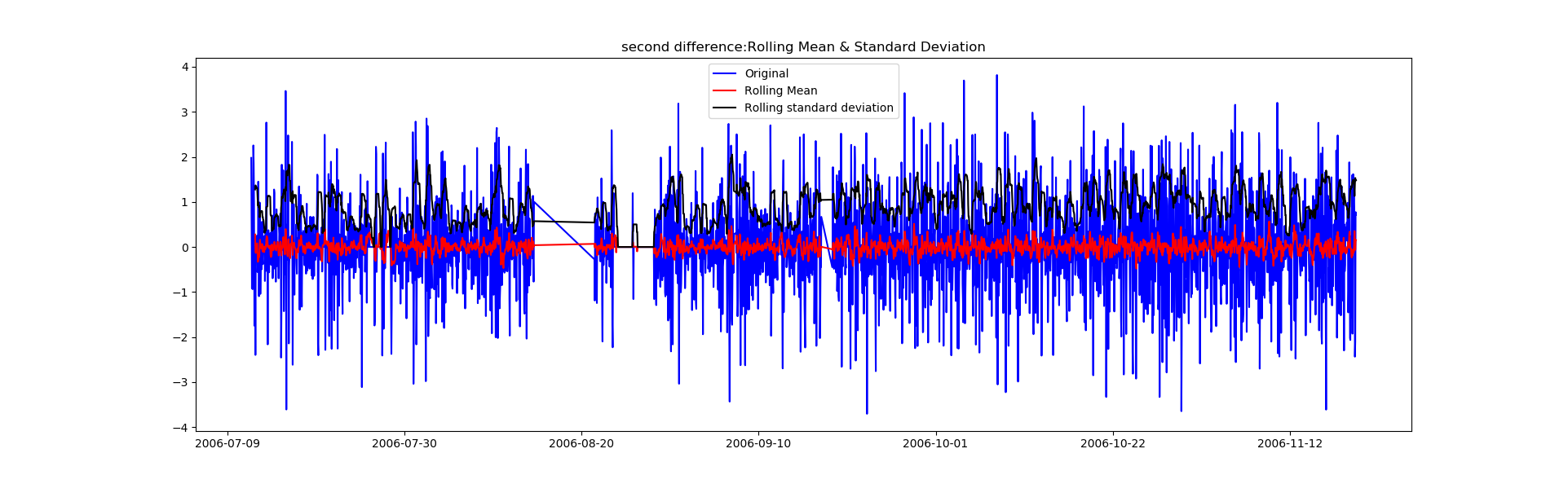


图4.4 Amazon QoS二阶差分序列及其均值序列、标准差序列

从图4.3、4.4中，可以明显看到，一阶差分、二阶差分过后的QoS序列的均值在时间轴上的振幅缩小了，对其进行ADF检验之后可知一阶差分、二阶差分序列具备良好的平稳性，可用于构建ARIMA模型。

4.3.2 识别ARIMA模型

数据平稳后，需要对ARIMA模型定阶，即确定p,q的阶数，本文通过计算序列的自相关函数（ACF）和偏自相关函数（PACF），然后根据表4.1的规则识别模型，其中，截尾是指结果值在某一阶后突然变得很小，如果ACF或PACF的结果在最初的d阶明显超过置信区间，而后几乎95%都落在置信区间，而且其衰减过程很突然，就称其为d阶截尾，而拖尾是指结果从某一阶后持续缓慢地由显著非零衰减为小值波动或者有超过5%的结果落在置信区间外。

表4.1 ARMA（p,q）模型识别规则

|  |  |  |
| --- | --- | --- |
| 模型 | 理论上的ACF | 理论上的PACF |
| AR(p) | 拖尾 | p阶截尾 |
| MA(q) | q阶截尾 | 拖尾 |
| ARMA(p,q) | 拖尾 | 拖尾 |

对Amazon QoS二阶差分序列构造ACF和PACF，结果图4.5所示：

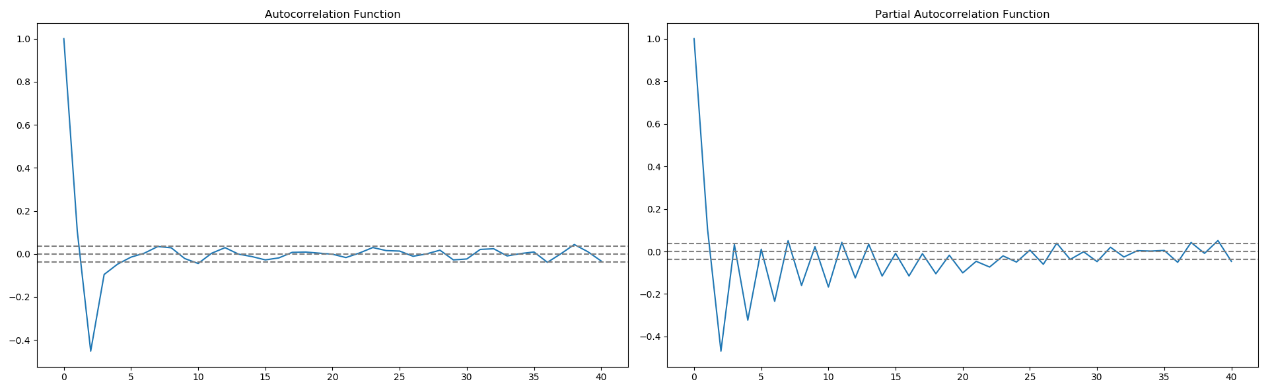


图4.5 二阶差分序列的ACF、PACF图

通过对图4.5的观察可知：ACF图在滞后2阶之后，自相关系数在置信区间内呈正弦波动，PACF图偏自相关系数从显著非零持续缓慢地缩减为小值波动。从图中可知，ACF图在[2,4]区间具有最大自相关性，因此q可取2,3,4, PACF图中由于时间序列的数据特性，尽管在区间[2,21]取得了最大的偏自相关性，但是具有拖尾性噪声，p可取[2,21]之间所有整数，然后根据均方根误差RMSE判断哪个模型的拟合度最好。RMSE定义为：

RMSE = （4.7）

其中N是样本大小，是t时刻观测值即真实值，是t时刻的预测值。经实验，模型ARIMA（18,2,3）的拟合效果最好。

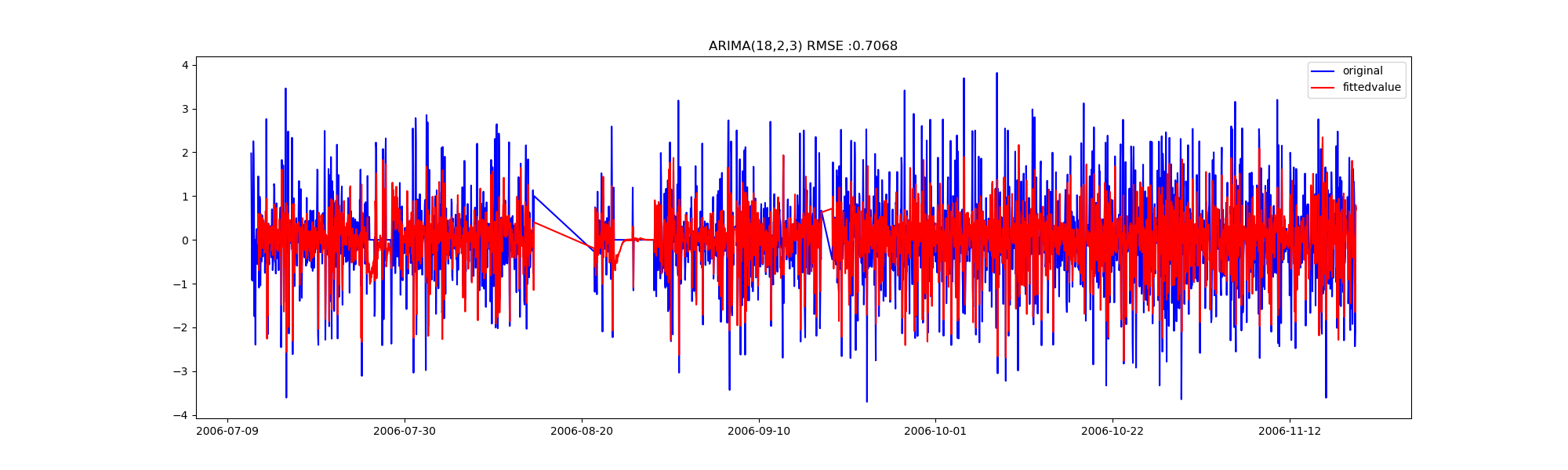


图4.6 ARIMA（18,2,3）拟合效果

由于图4.6的模型拟合值是对原数据进行二阶差分后的输入数据的拟合，所以需要对拟合值进行相应处理的逆操作，使得它回到与原数据一致的尺度。我们使用均方根误差RMSE和平均绝对误差MAE来评估模型样本内拟合的情况，RMSE定义在式4.7中已给出, MAE定义为：

MAE = （4.8）

其中N是样本大小，是t时刻观测值即真实值，是t时刻的预测值。原数据尺度下ARIMA（18,2,3）模型的拟合效果如图4.7所示：

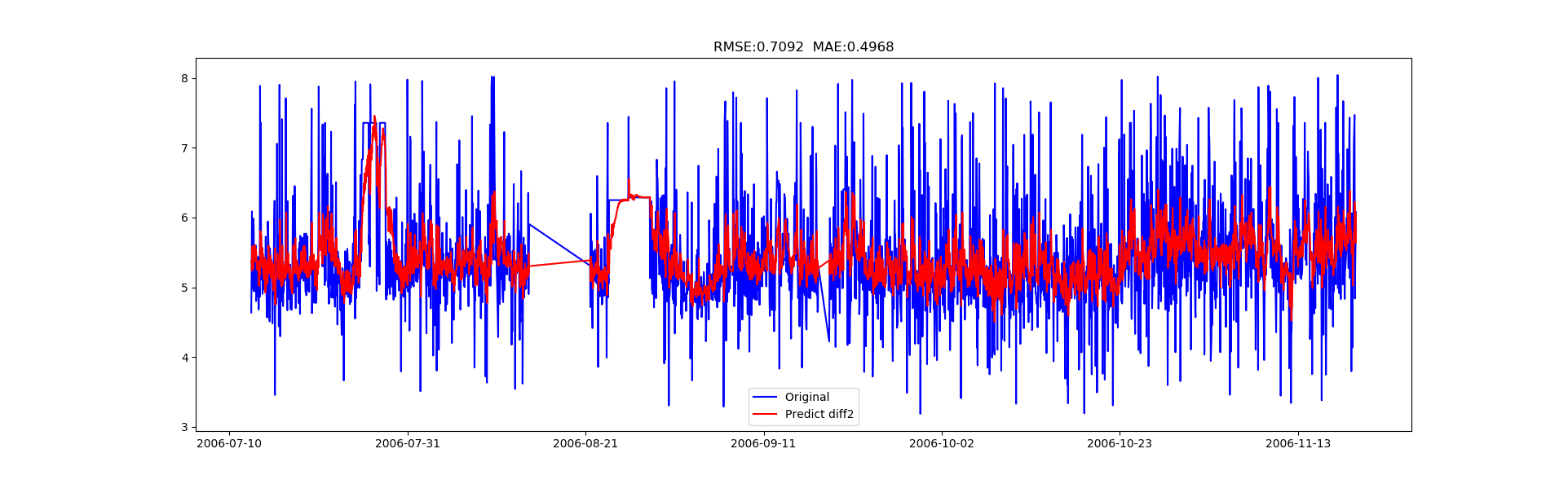


图4.7 原数据尺度下ARIMA（18,2,3）拟合效果

4.3.3 模型拟合评优

对于样本个数不多的情况，我们可以通过直接观察ACF图和PACF图来进行模型定阶，倘若我们要分析的时序数据量较大，又因为ARIMA模型不具有普遍性需要对不同的时间数据进行ARIMA模型识别，逐个定阶就显得不太合理，这时，我们可以依据一些模型拟合评优准则来选择最合适的模型。常用的评优标准如下：

（1）赤池信息准则（AIC）。AIC准则基于熵的概念，权衡估计模型复杂度与拟合优良性[36]，准则所用的统计量定义为：

AIC = 2k-2ln(L) （4.9）

其中k为参数个数，L是似然函数。

（2）贝叶斯信息准则（BIC）。构建模型时，随着参数个数增加，模型复杂度增加，但是可能会带来过拟合问题，BIC与AIC一样在评价模型时都引入了与模型参数数量相关的惩罚项，只是BIC的惩罚项更大，可以有效防止模型复杂度过高以及过拟合现象[36]。BIC的统计量定义为：

BIC = kln(n)-2ln(L) （4.10）

其中k和L的含义与式（4.9）相同，n为样本数量。

AIC和BIC准则都认为统计量越小表明模型越好，式（4.9）和式（4.10）中的k(参数个数)越小代表着模型越简洁，L越大表示模型越精确，所以AIC准则和BIC准则在评价模型时都既要求模型精确又考虑了简洁性，而BIC准则更有效防止了过拟合。

使用AIC准则和BIC准则自动定阶选择的最优模型分别为ARIMA（16,2,3）和ARIMA（3,2,2），拟合效果如下图所示：

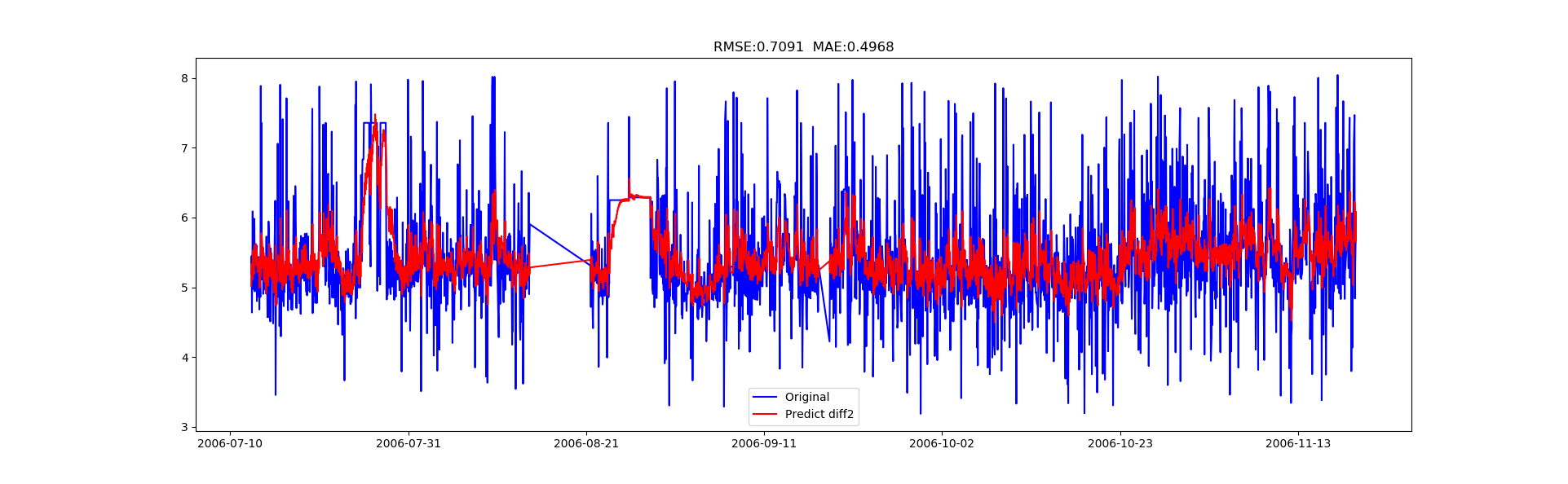


图 4.8 基于AIC准则最优模型： ARIMA（16,2,3）拟合效果

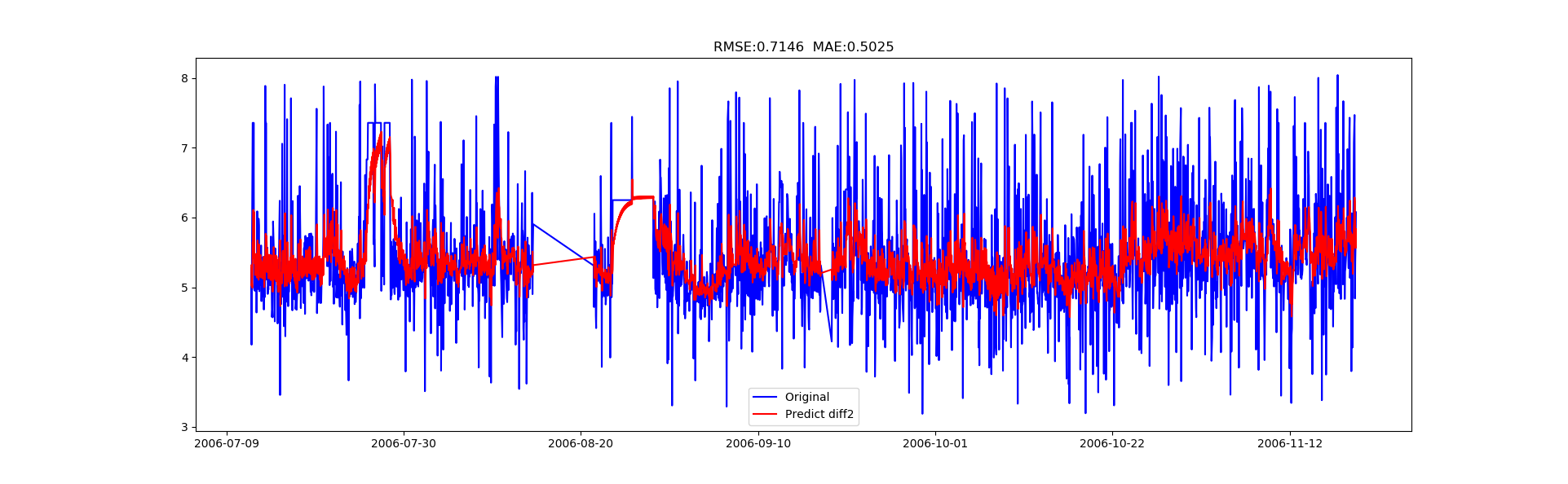


图4.9 基于BIC准则最优模型：ARIMA（3,2,2）拟合效果

ARIMA模型的定阶即取决于所用的评优标准，也极大依赖于实验人员的调参经验，由于具体问题中，时间数据具有不一样的特性，会对ARIMA模型的识别与最终预测效果造成很大的影响，需要实验人员具有丰富的实验经验和对样本数据特性进行充分分析，才能选择最好的模型。就Amazon QoS时序而言，模型ARIMA（18,2,3）和ARIMA（16,2,3）都是QoS时序预测模型的最佳选择。

4.4 本章小结

本章首先介绍了ARIMA模型的基本结构和建模步骤，然后实现了将ARIMA模型应用在QoS时序预测问题上。本章在阐述了ARIMA的基本理论之后，结合ARIMA模型对输入数据的要求和QoS时序数据的特性分析，构建了一个基于ARIMA模型的QoS时序预测方法，并进一步讨论了评优准则在模型自动定阶中的应用，从而得到最适合样本的预测模型。

# 5 基于LSTM的QoS时序预测方法研究

## 5.1 机器学习理论基础

Arthure Samuel将机器学习非正式定义为“在不直接针对问题进行编程的情况下，赋予计算机学习能力的一个研究领域”[37]，相较于这个比较抽象的定义，Tom Mitchell给出了一个更现代更正式的定义：“对于某类任务T和性能度量P，如果计算机程序在任务T上用P来衡量的性能随着经验E而自我完善，则称这个程序从经验E中学习”[38]。机器学习就是利用过往的经验，通过计算的手段，优化改善计算机程序自身的性能。在计算机程序中，经验表现为数据，将数据提供给机器学习算法，它就能从数据中学习到数据的类别或规律性等特征，然后建立能很好描述数据分布规律或类别属性的模型[39]，当面对新的情况时，模型会给我们提供相应的判断。神经网络是重要的机器学习算法之一，定义为是由具有适应性的神经元组成广泛并行互连的网络，它能够模拟生物神经系统对系统外界的交互做出相应的应答[40]。

### 5.1.1 神经元模型

神经元是构成神经网络的基本结构，如图4.1所示，一个神经元主要有三部分：输入与连接权重，激活函数，输出。一个神经元接收多个输入（,,…，），且每个输入上有一个权值,此外还有一个偏置项,这里我们将其当做是一个恒为“1”且连接权重为的输入信号。激活函数引入了非线性因素，提供了网络的非线性建模能力，常用的激活函数多是分段线性和具有指数形状的非线性函数，如sigmoid函数，tanh函数，relu函数等。神经元接收到的总输入值经过激活函数的变换，产生当前神经元的输出。一个神经网络就是由多个神经元按特定的层次结构组织连接起来的[40]。

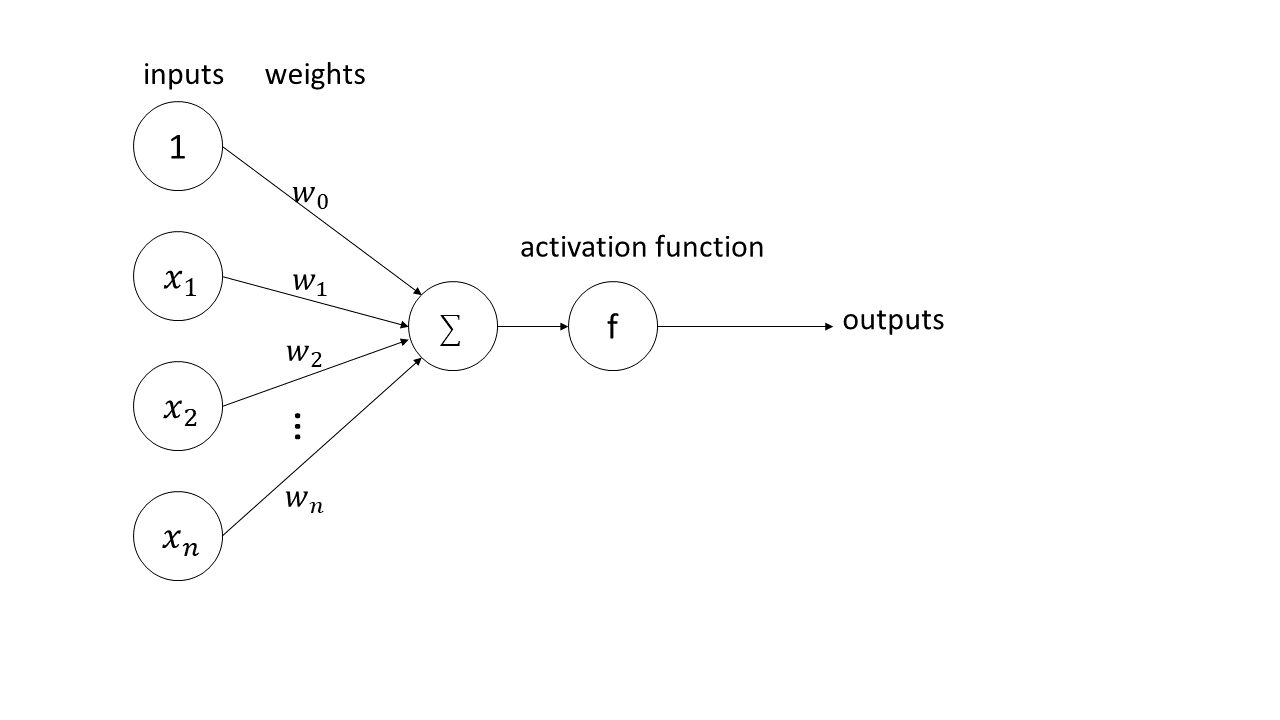


图5.1 神经元模型

### 5.1.2 目标函数与梯度下降法

目标函数，或称损失函数，是神经网络中的性能函数，目标函数越小，就代表模型拟合程度越好，常用的损失函数有均方误差MSE，绝对误差MAE等。目标函数取最小（最大）值时所对应的模型中的参数值，就是参数的最优值，很多时候我们只能获得目标函数的局部最小（最大）值，因此也只能得到模型参数的局部最优值。

目标函数要取最小（最大）值，其实就是优化问题，梯度下降法就是一种常用的优化算法。梯度是函数在某一点最大的方向导数[41]，函数沿着梯度方向具有最大的变化率，因此在优化函数时，沿着负梯度方向去减小函数值，就可以达到我们的优化目标。在最小化损失函数时，通过梯度下降法一步步迭代求解得到最小值就是在每一次迭代时计算当前的梯度，然后沿着梯度的负方向修改参数值，每次修改的步长称为学习速率，若太大则容易震荡而太小又收敛过慢，一次次迭代后就能取得最小值或最小值邻近值。

### 5.1.3 误差反向传播

神经网络其实就是多个神经元按照一定层次规则连接起来的[40]，具有负责接收输入信号的输入层，包含功能神经元的隐藏层和输出层，我们可以从输出层中获得模型的输出数据。如图5.2所示是一个全连接的神经网络,将输入信号{,，}提供给输入层神经元，然后将输入信息一层层地往前传，直到最后产生并输出结果{,}，网络训练的过程就是学习模型的参数（连接权重）的过程，模型计算误差然后进行权重的更新，而计算一个节点的误差项的时候，需要先计算每个与其相连的下一层节点的误差项，这就要求误差项的计算必须从输出层开始，然后逆向传播至隐含层，根据隐层神经元的误差来对参数进行调整，反向依次计算每个隐层的误差项并调整权重，直到与输入层相连的那个隐层，这就是误差反向传播的过程。

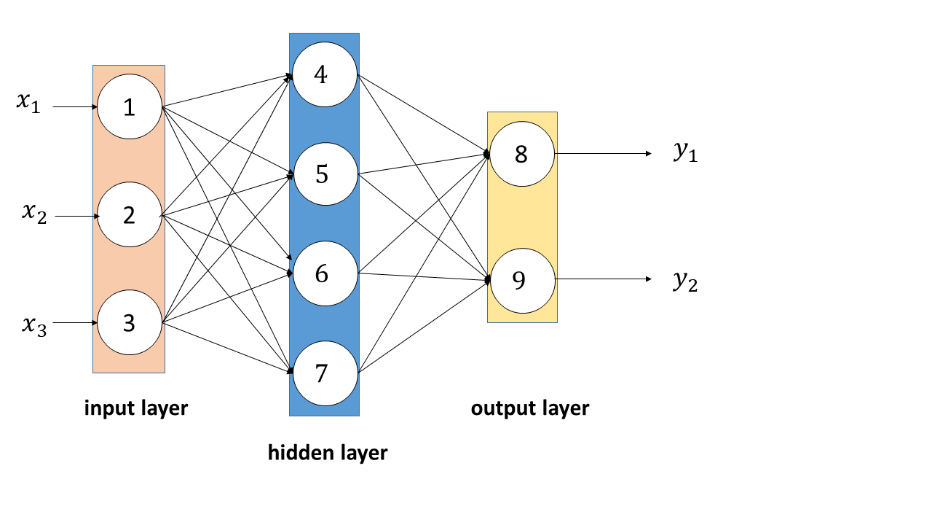


图5.2 全连接神经网络模型示例

## 5.2 循环神经网络模型结构

### 5.2.1 循环神经网络

循环神经网络(Recurrent Neural Networks,RNN) [42]是包含“循环”的网络——在处理单元之间既有内部的反馈连接又有前馈连接[43]，能够处理输入信号之间的时间前后关系，允许信息的持久化，因此常用于分析处理序列数据。时间序列就是一种特殊的序列数据，RNN是具有记忆的网络，可以从时间序列数据中学习到当前数据与过去数据的关系，然后利用学习到的信息将历史数据映射到未来数据，实现时序预测。循环神经网络的示例图如下：

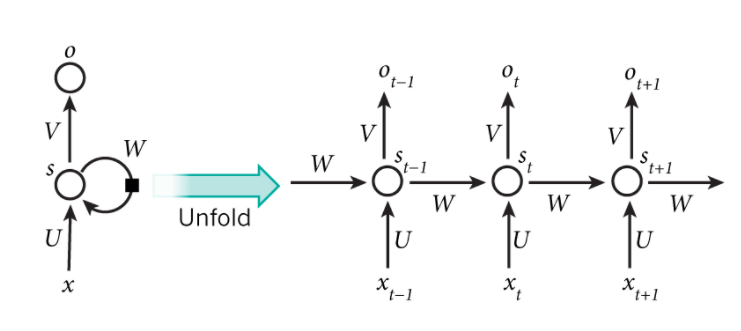
****

图5.3 循环神经网络示例图

其中x是输入层，o为输出层，s是隐藏层，U为输入层到隐含层的权重矩阵，V为隐含层到输出层的权重矩阵，循环神经网络的关键之处在于其隐藏层s的值不仅取决于输入层x,还取决于上一次隐藏层s的值，W即为上一次隐藏层s的值作为这一次的输入的权重矩阵。假设网络在t时刻接收输入,隐藏层为，且上一时刻的隐藏层为,则可以得到t时刻输出层的值为：

= g (V) （5.1）

= f (U) （5.2）

其中，g和f都是激活函数，结合图5.3中RNN网络展开模型示例可知，在循环神经网络中，输出值受前面按时间依次输入信号,，，，…影响。RNN最大的特点在于当前时刻的输出是由记忆和当前时刻的输出决定的，由于利用了历史信息，当任务涉及到与时序或与上下文相关时，RNN的性能要比其他人工神经网络的性能要好得多，因此循环神经网络相比其他神经网络模型更适合处理时序预测问题，它可以利用历史数据的时间依赖关系，来揭示数据特征，从而得到对未来的合理的预报。

### 5.2.2 长短时记忆网络

实践中，循环神经网络不能很好的处理较长的序列，这是因为RNN在训练中很容易发生梯度爆炸和梯度消失，导致训练时梯度不能在较长序列中一直传递下去，从而使模型无法捕捉到长距离的影响。长短时记忆网络（Long Short Term Memory Network,LSTM）的提出成功解决了原始循环神经网络难以处理长距离依赖的缺陷[44]。由于原始RNN隐藏层中的神经元只有一个状态（h）,它对短期输入非常敏感，而LSTM在原有神经元的基础上增加了一个状态（c）用来合理保存长期的状态，从而解决RNN难以处理长序列的问题。新增状态c称为记忆单元态，负责把记忆信息从序列的初始位置传递到序列的末端。LSTM利用控制门（gate）将原始RNN隐层神经元改造成一个可控记忆神经元，如图5.4所示：

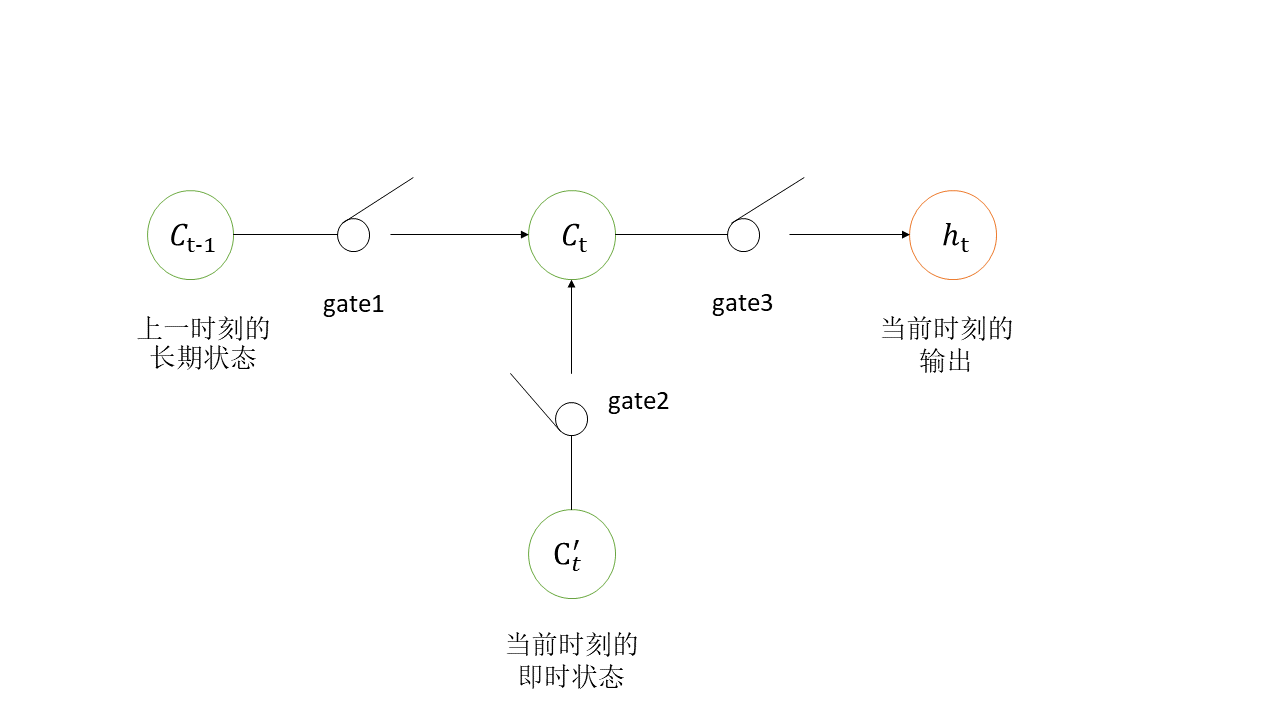


图 5.4 长期状态c的三个控制门结构

LSTM使用三个控制开关来控制长期状态，gate1负责决定把前一个长期记忆在多大程度上保存到中，它可以选择性地遗忘部分之前积累的信息， gate2控制把即时状态以多大程度加入到中，gate3控制是否把作为当前LSTM的输出。LSTM中的控制门（gate）是控制信息选择性通过，由一个sigmoid神经网络层和一个按位乘操作实现门结构的功能,sigmoid函数的定义如下：

（5.3）

sigmoid神经网络层输出0到1之间的数值，控制有多少量可以通过，特别地，0代表“不允许任何量通过”，1代表“允许任意量通过”。Gate1称为遗忘门，决定会从细胞状态中丢弃什么信息，遗忘门的结构由式（5.4）定义：

（5.4）

Gate2称为输入层，确定什么样的新信息被存放在细胞状态中,其结构如下定义：

（5.5）

确定要更新的信息后，就可以利用候选值来更新长期状态c

= \* + \* （5.6）

最后，gate3确定当前LSTM要输出什么值，其结构定义如下：

（5.7）

其中tanh函数取值在（-1,1）区间，定义如下：

（5.8）

是隐藏层状态，定义如下：

\* （5.9）

LSTM可以学习长期依赖信息的特点，使得它与RNN相比能更好地处理长序列，从历史数据中学习到更多需要的信息而舍弃一些不必要的信息，解决了RNN难以处理的长期依赖问题，更适合于处理QoS时间序列。

## 5.3 基于Keras框架实现的LSTM-QoS时序预测模型

### 5.3.1 数据预处理

在使用LSTM模型拟合QoS数据集之前，需要将数据处理为符合LSTM模型要求的规格，主要分为两个步骤：将时间序列转化为监督学习问题和将时间序列归一化。

#### 5.3.1.1 将时间序列转化为监督学习问题

LSTM模型训练是一个监督学习问题，通过从训练数据和前向验证积累的历史记录中获取最后几次观察结果并使用它来预测当前时间步来实现预测。监督学习问题由输入模式（X）和输出模式（Y）组成，并使用算法学习输入到输出的映射关系，目标是得到近似真实的映射，以便在有新的输入数据x输入时，可以预测该数据的输出变量y。时间序列是按时间索引排序的一系列数值，我们需要重构数据将其转化为监督学习问题，构建为如图5.5的输入模式X和输出模式Y。

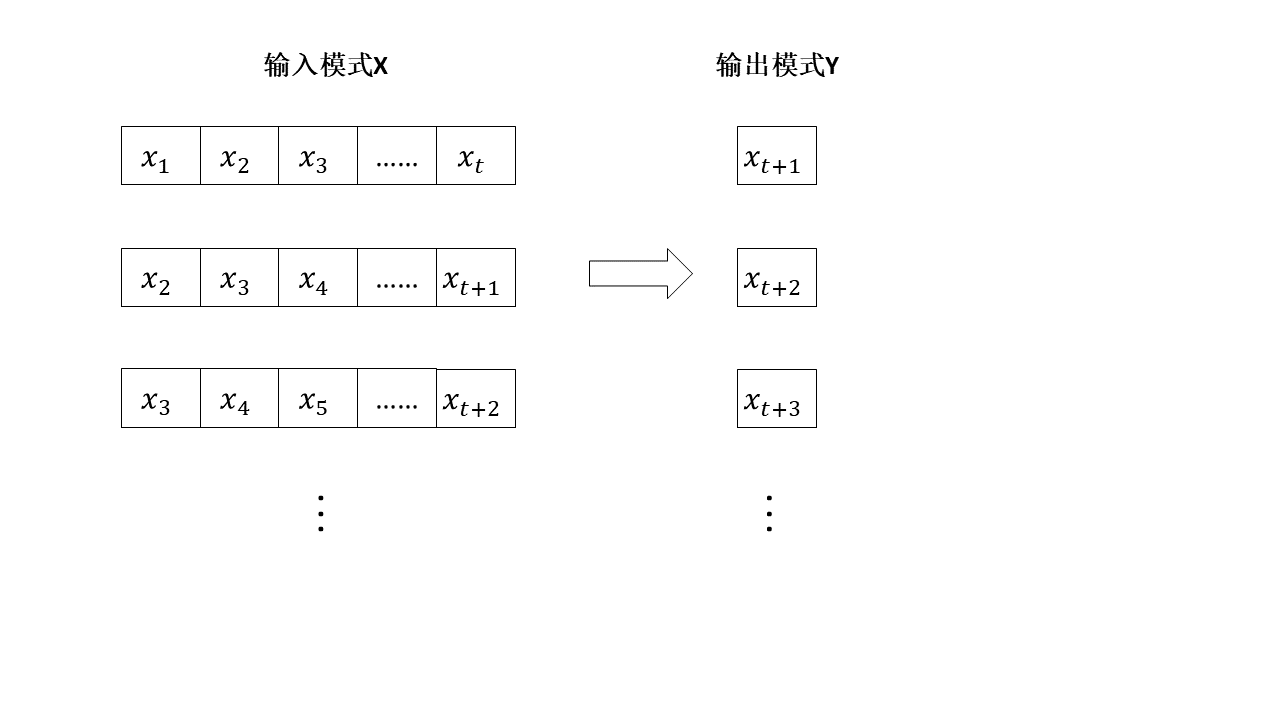


图 5.5 时间序列重构为监督学习问题

上图中，t代表timesteps,模型认为每个输入数据与前timesteps个陆续输入的数据有联系，即对当前值的预测需要通过向后看timesteps个数据学习数据分布规律，然后与已知的正确结果输出模式Y比较，进行模型更新，过程迭代地进行，学习在算法达到可接受的性能水平时停止。

#### 5.3.1.2 时间序列归一化

LSTM模型希望数据处于网络使用的激活函数的范围内，默认的激活函数是双曲正切tanh函数，它输出-1和1之间的值，这是时间序列数据的首选范围。为了使实验公平化，必须在训练集上计算缩放系数，并将其应用于缩放数据集和任何预测。这是为了避免使用来自测试数据集的知识来污染实验。本文实验中，定义了scale函数来实现将训练集归一化到区间（-1，1）之内，并将测试集和后续的预测按照训练集的缩放系数进行归一化，而进行模型拟合效果分析时我们使用invert\_scale函数来将数值反转到原始比例，以便进行预测精度分析。

### 5.3.2 利用Keras构建神经网络

Keras是基于theano/tensorflow的深度学习框架，支持快速实验，能够把设想迅速转换为结果，而避免那些复杂的数学运算。使用Keras搭建一个神经网络的步骤如下所示：

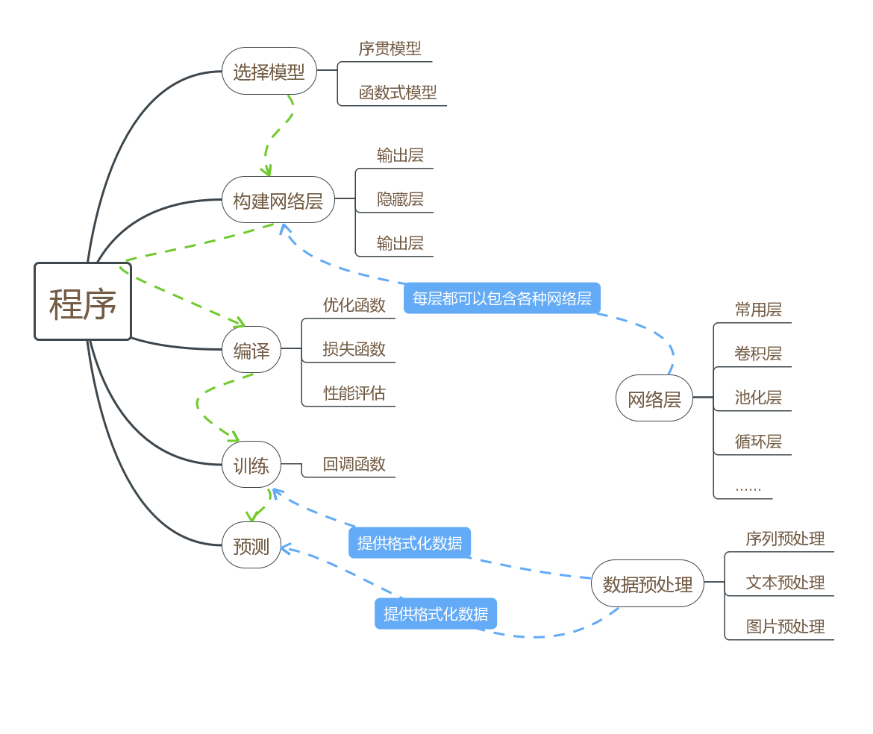


图 5.6 Keras框架下神经网络模型构建步骤

（1）创建模型。Keras有两种类型的模型：序贯模型和函数式模型，序贯模型是单输入单输出的模型，层与层之间只有相邻关系，没有跨层连接，函数式模型是多输入多输出的，层与层之间任意连接。

（2）构建网络层。神经网络的结构设计在网络层实现，网络的第一层需要指定input\_shape(samples,timesteps,features)。

（3）编译。在训练网络之前需要先调用编译过程来配置模型的学习过程，主要是指定损失函数和优化算法。

（4）训练。在给定输入数据和标签的训练集上迭代地进行预测并通过更新进行校正，学习在算法达到可接受的性能水平时或达到训练的总轮数时停止。

（5）预测。当模型适合于训练数据，它就可以用来进行预测。我们在所有训练数据中一次性拟合模型，然后根据测试数据逐个预测每个新时间步。

### 5.3.3 网络层结构设计

模型需要知道预期的输入形状（samples，timesteps，features），samples是来自域的独立观察结果，通常是数据行；timesteps就是每个输入数据与前timesteps个陆续输入的数据有联系；features是指在每个时间点上观察到的特征，这里我们将注意力集中在响应时间，features为1。本文实验中，我们堆叠3个LSTM层，使得模型能学习高级时序表征，前两层LSTM返回全部输出序列，但最后一层只返回输出序列最后一步，舍弃时序维度将输入序列转化为单一向量。

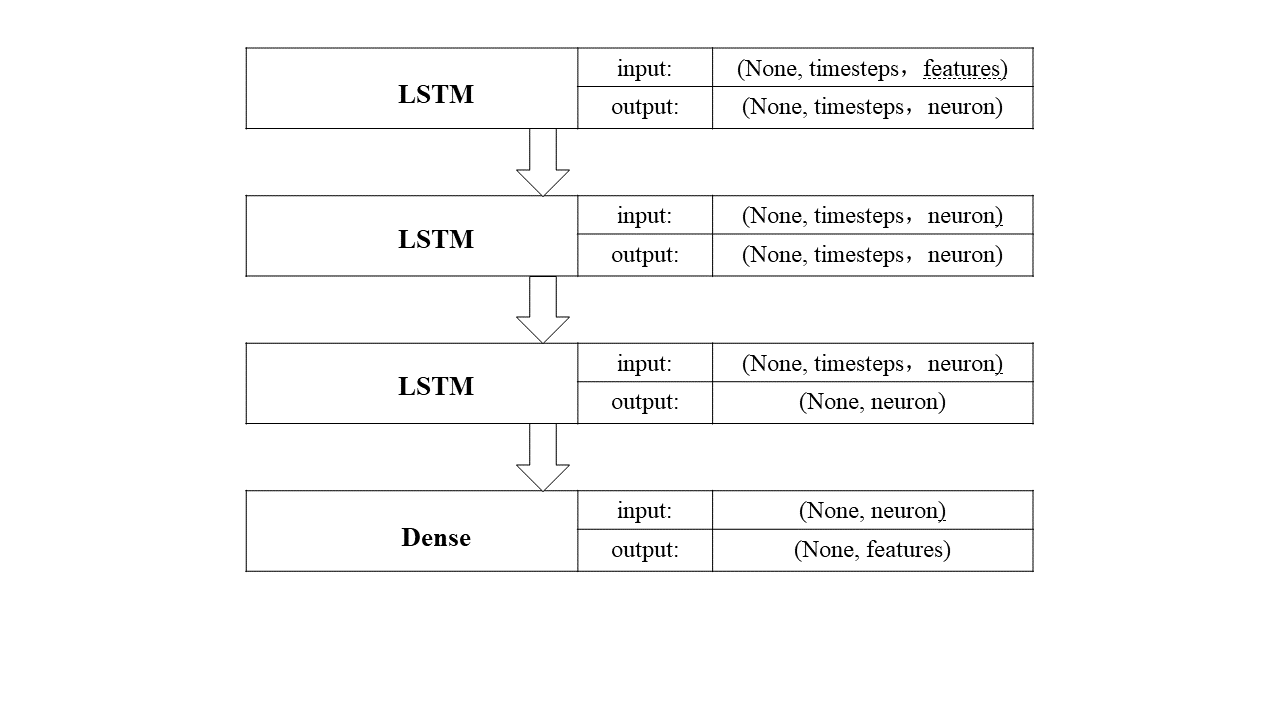


图 5.7 三层堆叠LSTM层网络结构

图5.7展示了一个堆叠三层LSTM的网络层结构，在第一层的时候需要提供给模型预期的输入形状（samples,timesteps,features）,而后面的层能够自动计算输入形状不需要额外提供。Neuron是神经元的数量也是当前层的输出维度，假如neuron=48,就一个时间点上的QoS观察值而言，LSTM内部简化看成:

（5.10）

该网络需要输出层中的单个神经元通过线性激活来预测下一个时间步中的QoS数值，所以第三层LSTM返回一个维度为neuron的向量并传给只有一个神经元且带有线性激活函数的全连接层来实现下一时间步的预测。随着模型训练的进行以及参数规模的增大，容易发生过拟合问题，因此在LSTM层之间加入Dropout层，在训练过程中每次更新参数时按一定概率随机断开输入神经元，以防止过拟合。

### 5.3.4 模型训练及调优模型

模型构建完成之后，在给定输入数据和标签的训练集上迭代地进行预测并通过更新进行校正，学习在算法达到可接受的性能水平或达到训练的总轮数时停止。LSTM网络的特点在于它可以学习和记忆长序列，在Keras中通过设置LSTM层中的“stateful”属性为真来实现，LSTM层维护一个batch\_size大小的数据之间的状态。Keras中LSTM层的参数更新是按批进行的，就是小批梯度下降算法，把数据分为若干组，称为batch，按批更新参数，这样，一个批中的一组数据共同决定了本次梯度的方向，一批数据中包含的样本数量称为batch\_size。小批梯度下降法对比每次更新参数都使用整个完整样本集的批梯度下降法计算速度更快，对比更新每一个参数都使用一个样本来更新的随机梯度下降法收敛性能更好。Batch\_size控制着更新网络权重的频率，选择合适的batch\_size对模型的好坏有着重要影响，batch\_size选取较大值时，模型训练的速度慢耗时长，但是能从更多数据上学习更新并改正，因此需要对其取值合理，兼顾训练速度和模型拟合效果。本文研究中根据数学经验，进行相关试验，选取batch\_size为128。

训练过程中当一个完整的数据集通过了神经网络一次并且返回了一次，这个过程称为一个epoch,网络会在每个epoch结束时报告关于模型学习进度的调试信息，由于LSTM网络是有状态的，我们必须控制内部状态何时重置，在一次epoch中批处理之间的LSTM层的状态必须保持，而在一次训练时期结束时需要重置内部状态，为下一次训练迭代做好准备。训练一个模型使其适合于训练数据然后能够对新的输入数据做出相应反应，需要将完整的数据集在神经网络中传递多次，也就是不仅仅更新权重一次或者说epoch个数远大于1。随着epoch数量增加，神经网络中的权重更新次数也增加，曲线从欠拟合变得过拟合。几个epoch才是合适的没有确定的答案，对于不同的数据集，答案是不一样的。针对本实验的QoS数据集，我们通过试验验证调试合适的epoch个数。以Amazon QoS时间序列为例，我们调试epoch个数。我们使用诊断方法研究模型配置，在诊断中，我们将生成模型在各个训练时期（epoch）预测水平的线图，通过研究该线图来理解给定配置的性能以及如何调整使其实现出更好的性能。

训练网络过程中，一个bacth中的样本规模大小与epoch个数一起通过影响更新权重的频率定义了网络学习数据的速度。所以在调试epoch个数的试验中我们保持固定的batch\_size，在每个epoch结束时分别使用训练数据集和测试数据集评测该模型，并记录均方根误差RMSE分数，试验运行完成后生成训练集和测试集的均方根误差RMSE分数序列图，如图5.8所示，训练集分数为蓝色，测试集分数为黄色。

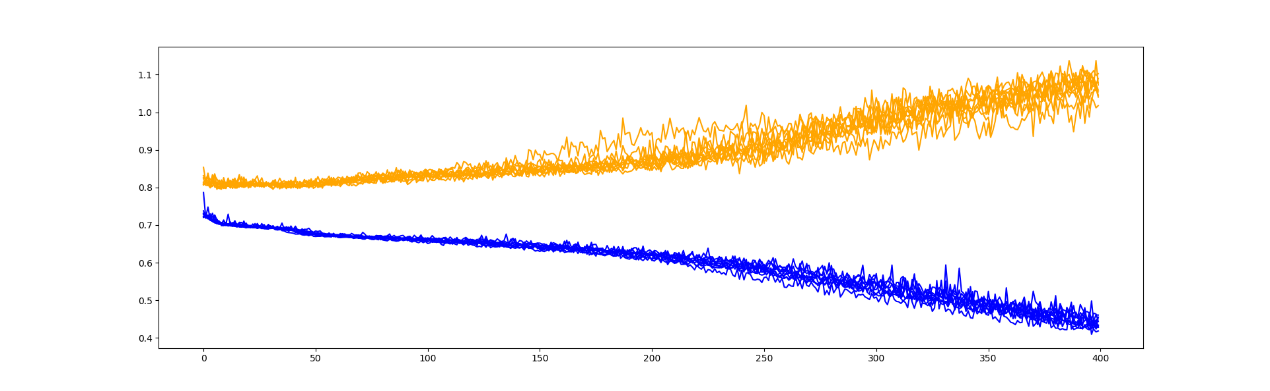


图 5.8 Amazon QoS时序各训练时段预测水平线图

试验结果清楚地表明，在试验运行中，随着训练epoch的增加，训练集的均方根误差呈下降趋势，这是个好迹象，表明了模型正在学习该预测问题并且已具备几分预测能力，而测试集的性能没有训练集的好，而且随着epoch的增加，测试集的均方根误差呈上升趋势，这可能是因为模型过渡拟合训练数据集，使得模型预测测试数据集的性能变差，我们通过调整Dropout层随机断开输入神经元的概率来防止模型过渡拟合训练数据集而对测试集性能不好，重新试验调试合适的epoch个数，综合考虑训练集拟合度和测试集的预测效果，最终选取在Amazon QoS时间序列上epoch为400。

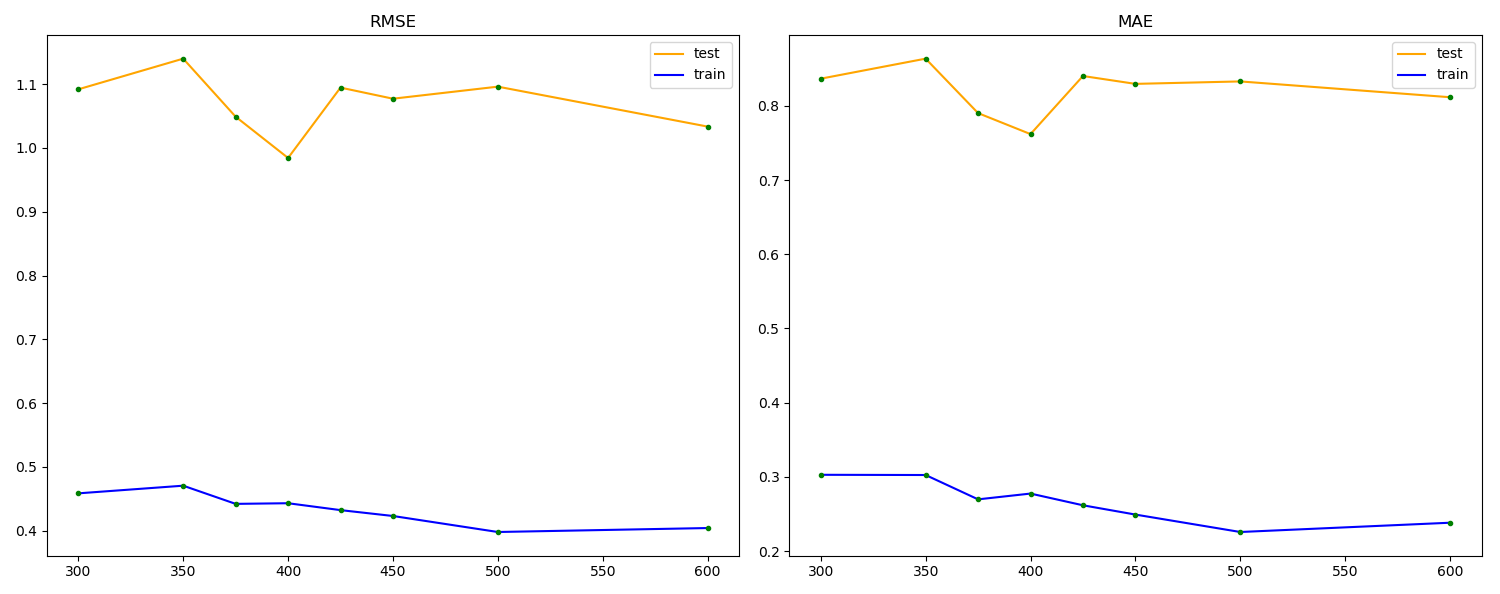


图 5.9 调整Dropout层后Amazon QoS时序训练时段预测水平线图

一旦模型适合于训练数据，它就可以用来进行预测。我们在所有训练数据中一次性拟合模型，然后根据测试数据逐个预测每个新时间步。以Amazon QoS时间序列为例，网络在训练集上训练完成后，对训练数据集进行一次拟合，利用均方根误差RMSE和平均绝对误差MAE来评估拟合效果，然后将测试数据的输入模式提供给模型逐个预测每个新时间步，并根据测试数据的标签对其预测效果进行评估，评价指标仍为RMSE和MAE。

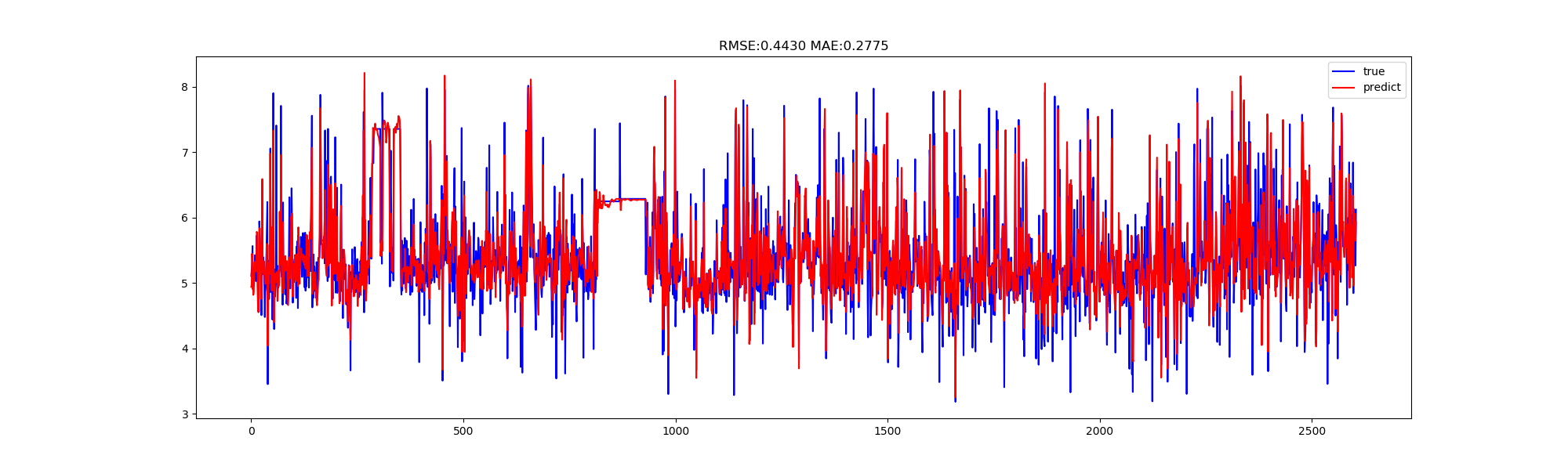


图 5.10 调优的LSTM模型在Amazon QoS数据测试集上的拟合效果

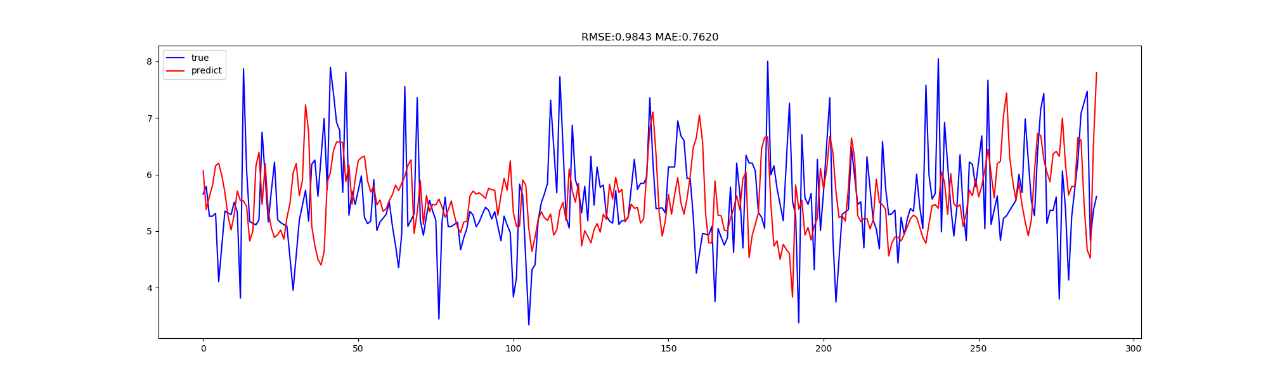


图 5.11 调优的LSTM模型在Amazon QoS数据训练集上的预测效果

## 5.4 本章小结

本章首先对机器学习的基本理论（神经元、梯度下降法、误差反向传播）进行了简单的叙述，然后进一步研究了适用于时序分析的神经网络——循环神经网络RNN及其变种长短时记忆网络LSTM的模型结构。基于此，根据QoS时间序列的数据特性，我们选取了能够处理长序列问题的LSTM模型，并基于Keras框架实现了LSTM-QoS时序预测模型，着重描述了如何进行数据预处理使得数据符合模型要求，模型的设计与构建，以及模型训练过程和如何调优模型，最后得到了适合于数据的网络，并进行训练集拟合和测试集预测评估。

# 6 实验结果分析与对比

## 6.1 实验设置

本文围绕时间感知的Web服务质量预测方法展开研究，实现了ARIMA-QoS时序预测模型和基于Keras框架的LSTM-QoS时序预测模型，本文设置了对比试验来进一步验证模型的预测效果：实现了简单移动平均预测方法、加权移动平均预测方法在本文构建的QoS时序数据集上的应用，通过对比评估标准来比较不同方法的预测误差和模型拟合程度。此外，作为结果分析的参考，我们还对四种预测方法的时间代价进行了统计。

我们使用均方根误差RMSE和平均绝对误差MAE来作为评估模型拟合程度的评价指标。均方根误差RMSE对一个观测值序列的特大（小）误差反应十分灵敏，所以它可以很好地反映预测的正确性，平均绝对误差MAE是每一次预测值与真实值的绝对偏差的平均值，它能够较好的反映预测的精度。针对每一组真实服务的QoS数据，如Amazon的QoS时序，我们分别将不同的预测方法应用于响应时间构成的QoS时序，对比预测结果的RMSE和MAE，评价模型的预测精度。

简单移动平均方法就是取前面已知的若干个样本的均值作为当前值的预测结果[45]，定义如下：

= （6.1）

其中，n为移动平均的时期个数，表示被预测时间点的前i个实际观察值。

简单移动平均方法简单，通常只对增长趋势不明显，波动变化比较平缓，并且与过去较为长远的时期联系不多的时序有效。

加权移动平均方法同样权衡当前被预测时刻过去的n个真实观察值的影响，但是对其效应赋予不同的权重，离当前值越近的观察值对预测影响越大，分配的权重也就越大，具体算法如下：

= （6.2）

其中表示当前值的前t-i个观察值对当前值的影响权重。

加权移动平均对比简单移动平均方法，认为近期数据包含着更多关于未来情况的信息[46]，分配越近的数据更大权重，这是合理的，也会提高预测的精度。但仍然无法处理时间序列具有趋势变化的情况，仅在时序没有明显趋势的时候能够准确反映实际情况。

图6.1、6.2分别为简单移动平均预测模型和加权移动平均模型在Amazon QoS时间序列上的预测表现。

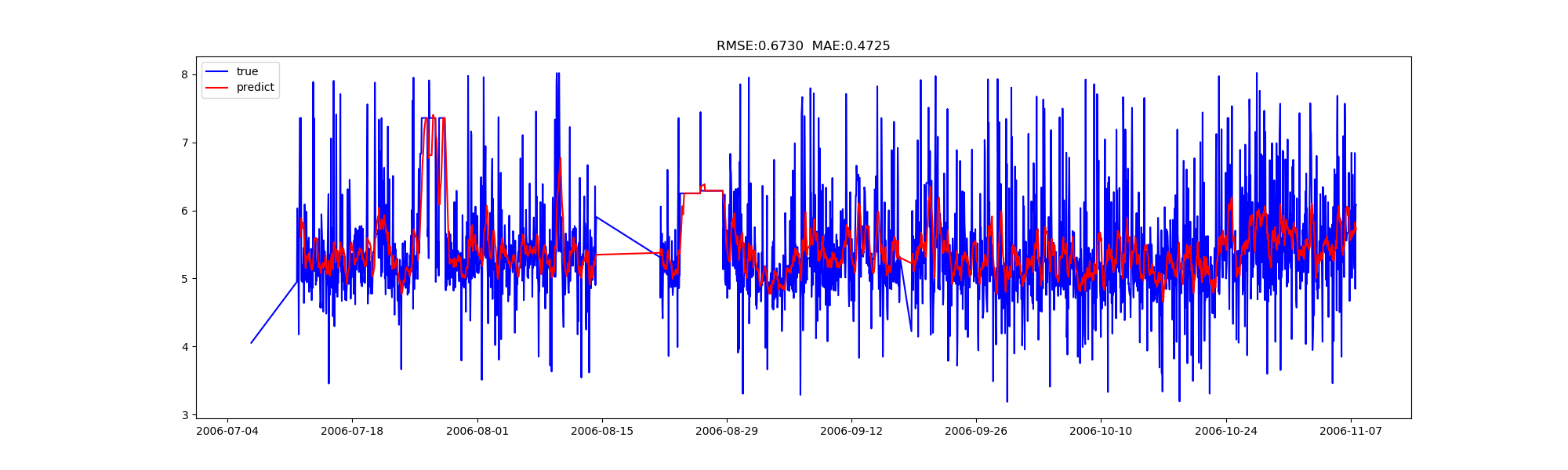


图 6.1 简单移动平均模型在Amazon QoS数据上的拟合效果

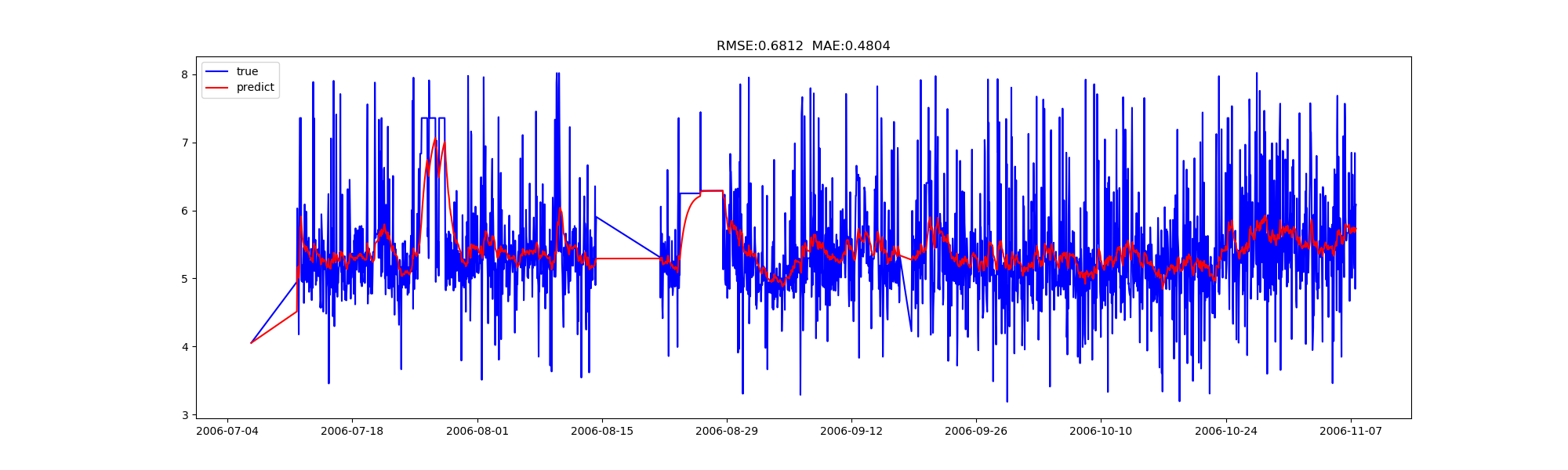


图 6.2 加权移动平均模型在Amazon QoS数据上的拟合效果

## 6.2 结果对比

针对十组实际服务的QoS数据，分别使用简单移动平均、加权移动平均、ARIMA模型、LSTM神经网络建模进行QoS时间序列预测学习。经过实证实验，各预测方法在训练集上预测结果的RMSE和MAE记录在表6.1，6.2中。

统计不同预测方法在十组QoS数据上进行模型训练、预测的总用时，并记录在表6.3中，实验环境为：

（1）操作系统：Windows 10

（2）处理器：Intel®Core™i5-4590 CPU @ 3.30GHz 3.30GHz 无GPU

表6.1 不同预测方法在十组QoS数据预测结果均方根误差RMSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务 | 简单移动平均 | 加权移动平均 | ARIMA模型 | LSTM模型 |
| Amazon | 0.6730 | 0.6812 | 0.6990 | 0.4850 |
| BLiquidity | 0.5509 | 0.5545 | 0.8067 | 0.5004 |
| Currency-  Converter | 0.2954 | 0.3012 | 0.2906 | 0.1916 |
| FastWeather | 1.0609 | 1.0785 | 1.0819 | 0.9725 |
| GetJoke | 0.3314 | 0.3314 | 0.3233 | 0.2763 |
| Google | 0.8422 | 0.8444 | 0.8134 | 0.7270 |
| Hyperlink-  Extractor | 0.7122 | 0.7231 | 0.7340 | 0.6358 |
| QuoteOfTheDay | 1.4461 | 1.3329 | 1.1648 | 0.8289 |
| StockQuotes | 0.1650 | 0.1646 | 0.1557 | 0.1295 |
| XMLDailyFact | 0.1892 | 0.1859 | 0.1821 | 0.1326 |

表6.2不同预测方法在十组QoS数据的预测结果平均绝对误差MAE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务 | 简单移动平均 | 加权移动平均 | ARIMA模型 | LSTM模型 |
| Amazon | 0.4725 | 0.4804 | 0.4878 | 0.3197 |
| BLiquidity | 0.3697 | 0.3819 | 0.5859 | 0.2874 |
| 50Currency-  Converter | 0.2102 | 0.2121 | 0.2060 | 0.1347 |
| FastWeather | 0.7676 | 0.8209 | 0.7962 | 0.7062 |
| GetJoke | 0.2420 | 0.2462 | 0.2339 | 0.1859 |
| Google | 0.5901 | 0.6131 | 0.5689 | 0.5200 |
| Hyperlink-  Extractor | 0.5094 | 0.5348 | 0.5297 | 0.4476 |
| QuoteOfTheDay | 1.2051 | 1.1210 | 0.8297 | 0.5474 |
| StockQuotes | 0.1065 | 0.1119 | 0.0981 | 0.0750 |
| XMLDailyFact | 0.1437 | 0.1434 | 0.1345 | 0.0907 |

表6.3不同预测方法的时间代价（单位：秒/s）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务 | 简单移动平均 | 加权移动平均 | ARIMA模型 | LSTM模型 |
| Amazon | 0.3690 | 0.3948 | 216.1639 | 640.6782 |
| BLiquidity | 0.3532 | 0.3553 | 176.2174 | 678.5373 |
| 50Currency-  Converter | 0.2552 | 0.2512 | 157.7434 | 579.9501 |
| FastWeather | 0.3012 | 0.2992 | 92.7605 | 381.3663 |
| GetJoke | 0.3142 | 0.3092 | 242.4672 | 630.9583 |
| Google | 0.3272 | 0.3162 | 273.1119 | 744.2766 |
| Hyperlink-  Extractor | 0.3683 | 0.3613 | 168.2538 | 771.7814 |
| QuoteOfTheDay | 0.3673 | 0.3423 | 171.4020 | 783.0299 |
| StockQuotes | 0.3182 | 0.3052 | 161.9083 | 799.9588 |
| XMLDailyFact | 0.3182 | 0.3132 | 127.5351 | 674.2651 |

可以通过观测十组QoS数据应用不同预测方法的RMSE和MAE的点线图更直观的比较不同预测方法的预测效果。

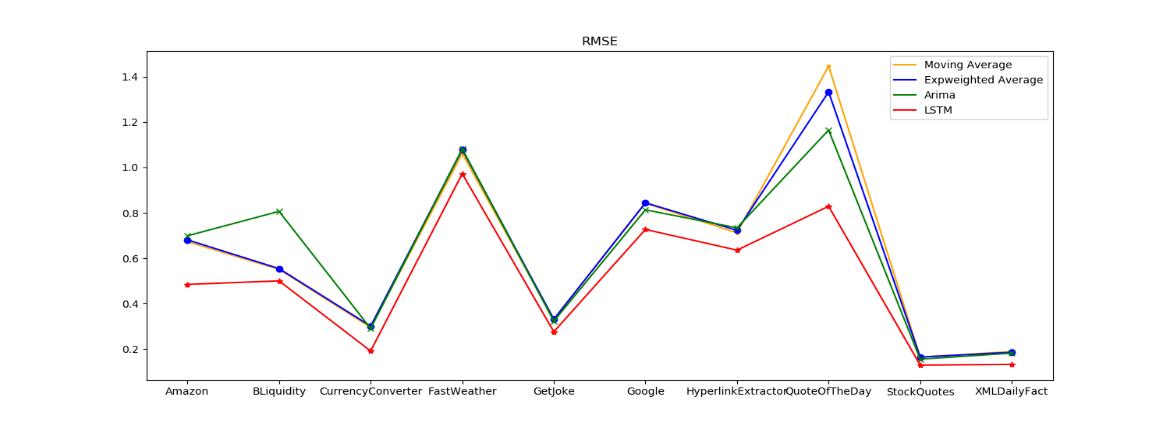


图 6.3 不同预测方法在十组QoS数据的预测结果均方根误差

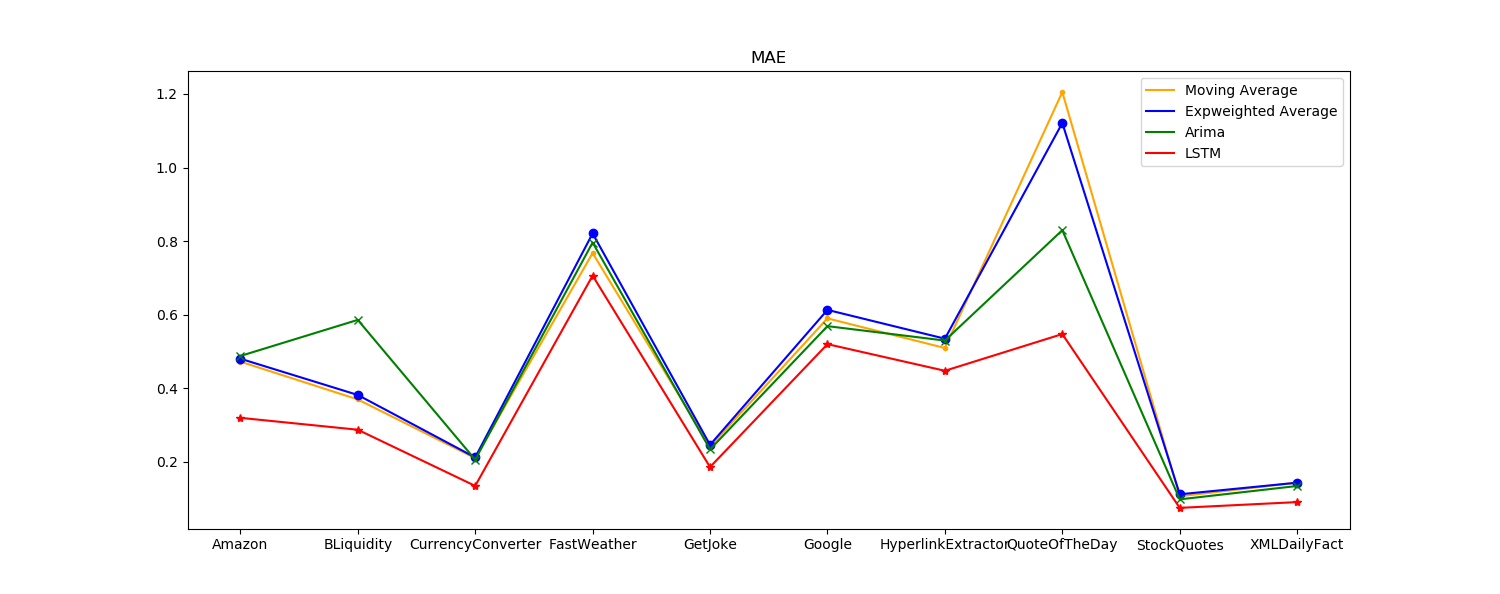


图 6.4 不同预测方法在十组QoS数据的预测结果平均绝对误差

从表6.1、6.2和图6.3、6.4可以看出，不论是从RMSE还是MAE的角度衡量，基于ARIMA模型的QoS预测方法在大部分数据上的表现稍好于简单移动平均和加权移动平均，而在个别QoS数据上预测效果劣于这两种简单模型，这是因为ARIMA预测模型对数据具有周期性等要求，当数据的周期性不明显时，ARIMA对数据的适用性很低。基于机器学习的LSTM-QoS时序预测模型在十组QoS数据上的预测精度都要高于其他三个预测方法，证明了LSTM-QoS预测模型在这十组QoS样本数据的预测任务中表现出了最佳性能，预测准确度优于简单移动平均方法、加权移动平均方法以及基于ARIMA模型的QoS预测方法。如表6.3所示，LSTM-QoS模型和ARIMA-QoS模型都是以时间为代价换取预测精度的提升，虽然LSTM-QoS模型的用时远超过简单移动平均、加权移动平均方法，但是模型计算耗时仍在可接受范围内。

## 6.3 本章小结

本章通过对比实验，将简单移动平均方法、加权移动平均方法、与本文实现的ARIMA-QoS时序预测模型、LSTM-QoS时序预测方法分别应用于十组QoS数据，利用预测结果的均方根误差和平均绝对误差来评价预测方法的预测精度，实验证明LSTM-QoS时序预测方法要优于其他三个方法，在QoS预测问题有较好的表现。

# 7 总结与展望

## 7.1 总结

本文围绕时间响应的Web服务质量预测方法展开研究，主要针对QoS数据预测进行了讨论，实现了基于ARIMA模型的QoS预测模型、基于Keras框架的LSTM-QoS时序预测模型，并将预测效果与简单移动平均预测方法和加权移动平均预测方法的预测效果进行了对比分析。主要研究工作和研究成果主要有以下几个方面：

（1）分析十组实际服务的QoS数据结构与时序特征，分解QoS序列得到随机噪声项，分析周期性和趋势性，使用ADF单位根检验和时序图观察验证平稳性，构建了QoS时间序列并进行数据预处理，为课题后续的实验提供了一套完整有效的时序数据集。

（2）基于QoS数据预测问题开展ARIMA模型方法的研究与实现，根据序列的特征选择适当的拟合模型，检验模型，优化模型，评估模型在QoS预测问题上的表现。

（3）基于机器学习和循环神经网络模型结构的学习开展了长短时记忆网络LSTM模型应用于QoS时间序列预测问题上的研究，实现了基于Keras框架的LSTM-QoS时序预测模型。对时间序列进行预处理使其符合模型输入数据的要求，设计构建模型，训练网络调优模型，网络在QoS数据预测任务上取得了较好的预测效果。

（4）利用均方根误差RMSE和平均绝对误差MAE来作为评估模型拟合程度的评价指标，将简单移动平均方法、加权移动平均方法、ARIMA-QoS时序模型及LSTM-QoS时序模型应用于本文构建的QoS时间序列数据，，对比分析四种不同的方法在十组不同服务的QoS数据集上的表现，综合分析，LSTM-QoS预测模型在QoS预测任务上具有更高的预测精度。

## 7.2 展望

随着Web服务数量的逐年增长，QoS准确预测在Web服务推荐系统和面向服务的体系结构（SOA）服务组合中的应用需求越来越大，不论在科学研究领域还是工业领域，有效的QoS预测具有重大的研究意义和实用价值。因为QoS数据的特有性质，在QoS预测方面具有一定限制，导致了各个模型的缺陷难以避免：

（1）基于Arima的QoS预测模型不具有普遍性，在部分数据上表现不如简单模型，这是因为Arima模型对数据具有一定的要求，模型在具有明显周期性的数据上有比较好的表现。

（2）基于机器学习的QoS预测模型相对于简单模型在预测误差方面的好处是显著的，但是在训练模型上耗时较长，时间代价大，且模型参数调试过程需要大量试验和实验经验。

（3）基于机器学习的QoS预测模型在训练数据集上的预测精度大大高于测试数据集，当时间序列显示出异常值时很难进行准确的预测。

基于QoS预测问题的研究，针对以上不足，未来工作旨在对LSTM-QoS预测模型进行优化，以期达到更好的预测效果：

（1）基于多元时间序列的LSTM-QoS预测模型，建模具有相关性的多个属性，可能提高模型对QoS数据的预测精度。

（2）模型的输入模式增加滞后观察或错误序列作为附加输入特征，需要进行实验来验证附加输入特征能否提高模型预测精度。

此外，模型的优化目标除了进一步提高预测准确性，还应该综合考虑普遍适应性和建模成本，未来可以考虑做进一步优化提升，使得LSTM-QoS时序预测方法具有更高的精确性和更低的建模成本。

# 参考文献

1. Laplante P A, Zhang J, Voas J. Services Computing[J]. Springer & Tsinghua University Press, 2007.
2. 王磊. 基于服务组合的“系统的系统”的可靠性时间序列预测方法及关键技术研究[D]. 东南大学, 2017.
3. 蓝岩华. WEB服务协同工作功能测试方法的实现[M].
4. George E P Box, Gwilym M.Jenkins. 顾岚主译. 时间序列分析: 预测与控制[M].北京：中国统计出版社, 1997, 9: 1-16.
5. 张建辉. 基于BP神经网络的时序预测模型的研究[D]. 太原理工大学, 2017.
6. Yu T, Lin K J. Service selection algorithms for Web services with end-to-end QoS constraints[C]// e-Commerce Technology, 2004. CEC 2004. Proceedings. IEEE International Conference on. IEEE, 2004:129-136.
7. 申利民，陈 真，李 峰.一种考虑QoS数据不确定性的服务选取方法[J]. 计算机集成制造系统，2013, 19（10）：2652-2663
8. Hadad J E, Manouvrier M, Rukoz M. TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition[J]. IEEE Transactions on Services Computing, 2010, 3(1): 73-85.
9. 申利民,陈真,李峰.基于时序分析的Web服务QoS协同预测[J].小型微型计算机系统,2015,36(09):1932-1938.
10. Engle R F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation[J]. Econometrica, 1982, 50(4):987-1007.
11. Chen J H, Kuo T W, Lee A C. Run-by-Run Process Control of Metal Sputter Deposition: Combining Time Series and Extended Kalman Filter[J]. IEEE Transactions on Semiconductor Manufacturing, 2007, 20(3):278-285.
12. 施闻明，徐彬，短记忆性时间序列的建模分析及预测[J] 计算机仿真,2007.
13. Chen A S, Leung M T. Regression neural network for error correction in foreign exchange forecasting and trading[J]. Computers & Operations Research, 2004, 31(7):1049-1068.
14. Kwon Y K, Moon B R. A Hybrid Neurogenetic Approach for Stock Forecasting [M]. IEEE Press, 2007.2007, 18(3): 851-864.
15. Cao L J, Tay F H. Support vector machine with adaptive parameters in financial time series forecasting[J]. IEEE Transactions on Neural Networks, 2003, 14(6): 1506-18.
16. Wang J Z, Wang J J, Zhang Z G, et al. Forecasting stock indices with back propagation neural network[J]. Expert Systems with Applications, 2011, 38(11): 14346-14355.
17. Naeini M P, Taremian H, Hashemi H B. Stock market value prediction using neural networks[C]// International Conference on Computer Information Systems and Industrial Management Applications. IEEE, 2010:132-136.
18. Yoo P D, Kim M H, Jan T. Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation[C]// International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol. IEEE Computer Society, 2005: 835-841.
19. Shao L, Zhang J, Wei Y, et al. Personalized QoS Prediction forWeb Services via Collaborative Filtering[C]// IEEE International Conference on Web Services. IEEE Computer Society, 2007:439-446.
20. Zheng Z, Ma H, Lyu M R, et al. QoS-Aware Web Service Recommendation by Collaborative Filtering[J]. IEEE Transactions on Services Computing, 2011, 4(2): 140-152.
21. Chen L, Yang J, Zhang L. Time Based QoS Modeling and Prediction for Web Services[C]// International Conference on Service-Oriented Computing. Springer Berlin Heidelberg, 2011:532-540.
22. Hua Z, Meng L I, Zhao J, et al. Web Service QoS Prediction Method Based on Time Series Analysis[J]. Journal of Frontiers of Computer Science & Technology, 2013, 7(3): 218-226.
23. Cavallo B, Penta M D, Canfora G. An empirical comparison of methods to support QoS-aware service selection[C]// 2010:64-70.
24. 王燕. 应用时间序列分析-第3 版[M]. 中国人民大学出版社, 2012.
25. 何书元. 应用时间序列分析[M]. 北京大学出版社, 2004.
26. 刘超. 中国证券投资基金系统制度研究[D]. 天津大学, 2009.
27. 邓祖新. 数据分析方法和SAS系统[M]. 上海财经大学出版社, 2006.
28. 聂巧平, 张晓峒. ADF单位根检验中联合检验F统计量研究[J]. 统计研究, 2007, 24(2):73-80.
29. 金玉国. 计量经济学 : Econometrics[M]. 经济科学出版社, 2015.
30. 李文, 刘永辉, 杨娇. 基于R语言的时间序列平稳性检验及实证分析——以中国宏观经济变量为研究对象[J]. 中国商论, 2017(16):150-153.
31. 张朝雄, 沈旻, 张慧. 缺失数据插补方法比较研究[J]. 市场研究, 2007(9):33-35.
32. 李金林, 赵中秋, 马宝龙. 管理统计学.第2版[M]. 清华大学出版社,2011.
33. 王怀亮. 箱须图在识别统计数据异常值中的作用及R语言实现[J]. 商业经济, 2011(5):64-65.
34. ACHarvey, P.H.J.Todd. Forecasting Economic Time Series With Structural and Box-Jenkins Models: A Case Study[J]. Journal of Business & Economic Statistics, 1983, 1(4): 299-307
35. 张小斐, 田金方. 基于ARIMA模型的短时序预测模型研究与应用[J]. 统计教育, 2006(10):7-9.
36. Burnham K P, Anderson D R. Multimodel Inference Understanding AIC and BIC in Model Selection[C]// Amsterdam Workshop on Model Selection August. 2004:93.
37. Samuel A L. Some studies in machine learning using the game of checkers[M]// Computers & thought. MIT Press, 1995:206-226.
38. Mitchell T M. Machine learning (mcgraw-hill[J]. Machine Learning, 1997.
39. 朱成璋.基于机器学习的时间序列预测关键技术研究[D].国防科学技术大学,2014.
40. 周志华. 机器学习 : = Machine learning[M]. 清华大学出版社, 2016.
41. 扈志明. 微积分教程(下册)[M]. 清华大学出版社, 2000.
42. Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proc Natl Acad Sci U S A, 1982, 79(8):2554-2558.
43. Elman J L. Finding structure in time-Cognitive Science, 1990, 14(2): 179-211.
44. Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
45. 吴清烈, 蒋尚华. 预测与决策分析[M]. 东南大学出版社, 2004.
46. 暴奉贤陈宏立. 经济预测与决策方法[M]. 暨南大学出版社, 2002.

# 致 谢

时光荏苒，大学生活即将结束，四年的学习生活使我受益匪浅。毕业离校之期将近，毕业论文的工作也随之进入了尾声，值此之际，我要向所有期间给予我支持、帮助和鼓励的人表示我最诚挚的谢意。

首先，我要衷心感谢我的指导老师李兵老师对我的指导，从论文的选题到完成，从理论框架到具体实现，李老师一直悉心指导。李老师以严谨的治学之道、宽厚仁慈的胸怀、诲人不倦的高尚师德、谦虚和善的学者风范，为我树立了一辈子学习的典范，他的教诲与鞭策激励着我在学习道路上继续励精图治，开拓创新。同样感谢课题组的王健老师，在本文完成过程中提出的宝贵的意见和建议，还要感谢实验室的师兄师姐以及同学们，他们在课题的实现过程中给予我无私的帮助。

其次，我要感谢我的辅导员赵玉琦老师，感谢赵老师在大学四年学习生活中给了我悉心的教诲和热情的关怀与鼓励。正是赵老师鼓励我们不局限于课堂的多方面尝试，让我们收获了一个丰富充实的大学生活，从他身上我们学到了如何求知治学如何为人处事，感谢他在学习上给我以指导，在思想生活上给我以关心与帮助。

同时，感谢和我一起生活了四年的舍友，四年来，我们朝夕相处，共同进步，感谢你们给予我的所有关心和帮忙。感谢我亲爱的朋友们，是你们在我失意时给我鼓励，在我失落时给我支持，感谢你们陪我一路走来，分享欢乐与泪水。

最后，感谢父母家人在求学生涯中给予我无微不至的关怀与照顾，一如既往地支持与鼓励，你们的爱与支持是我前进的最大动力。