**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**学士学位论文**

**BACHELOR THESIS**



论文题目 **PM2.5污染实时预测及其在移动设备中的设计与实现**

学 院  **计算机科学与工程学院网络空间安全学院**

专 业  **信息安全**

学 号 **2015060204001**

作者姓名 **孟华彬**

指导 **付波**

**摘 要**

近年来，随着现代化工业的发展和世界人口数量的增涨，大气环境在不断受到污染，大气里的细颗粒物(PM2.5)和其他空气污染物在空气的含量占比显著增加,空气污染仍然是中国乃至全世界需要重视的问题,特别是我国污染严重的北方地区。为了研究PM2.5值的发展走势及对其进行预测，本论文基于2018年收集部分几个月以每天为间隔的PM2.5浓度数据,使用机器学习方法预测成都市PM2.5的浓度水平。本课题中使用了时间序列数据分析预测方法中的ARIMA算法（差分整合移动平均自回归模型）对PM2.5进行建模并预测。为了构建PM2.5预测模型,我们通过Pandas对数据进行预处理处理。将预处理后的数据进行模型训练。然后通过Android Studio对手机APP进行开发，服务器搭建采用pycharm+Django搭建，然后把训练好的模型放到服务器上运行。这样当需要对PM2.5预测时，直接通过手机APP进行对服务器的访问就能获取未来时间PM2.5的预测值。这项课题研究的意义在于通过实现对PM2.5的分析和预测,可以及时地告知人们并使其采取适当措施,在一定程度上避免影响他们的健康

**关键词：**大气污染，PM2.5，时间序列数据分析算法，ARIMA算法，PM2.5预测

ABSTRACT

In recent years, with the development of modern industry and the increase of the world population, the atmospheric environment is continuously polluted, and the proportion of fine particulate matter (PM2.5) and other air pollutants in the air is significantly increased. It is still a problem that China and the world need to pay attention to, especially in the northern regions where China is seriously polluted. In order to study the development trend of PM2.5 value and predict it, this paper is based on the collection of PM2.5 concentration data at intervals of several months in 2018, using machine learning to predict the concentration level of PM2.5 in Chengdu. In this paper, the ARIMA algorithm (differential integrated moving average autoregressive model) in the time series data analysis and prediction method is used to model and predict PM2.5. To build the PM2.5 predictive model, we preprocessed the data through Pandas. The pre-processed data is model trained. Then develop the mobile app through Android Studio, build the server with pycharm+Django, and then put the trained model on the server. In this way, when the PM2.5 prediction is needed, the server's access to the server directly can obtain the predicted value of PM2.5 in the future time. The significance of this research is that by analyzing and predicting PM2.5, people can be informed in a timely manner and appropriate measures can be taken to avoid affecting their health to a certain extent.

**Keywords:**  air pollution, PM2.5, time series data analysis algorithm,

ARIMA algorithm, PM2.5 prediction

目 录

[第一章 绪 论](#_Toc16639_WPSOffice_Level1) [1](#_Toc16639_WPSOffice_Level1)

[1.1 课题研究的背景与意义](#_Toc2868_WPSOffice_Level2) [1](#_Toc2868_WPSOffice_Level2)

[1.1.1 课题研究的背景](#_Toc2868_WPSOffice_Level3) [1](#_Toc2868_WPSOffice_Level3)

[1.1.2 课题研究意义](#_Toc17397_WPSOffice_Level3) [1](#_Toc17397_WPSOffice_Level3)

[1.2 PM2.5的国内外研究历史与现状](#_Toc17397_WPSOffice_Level2) [1](#_Toc17397_WPSOffice_Level2)

[1.2.1国内研究现状](#_Toc8407_WPSOffice_Level3) [1](#_Toc8407_WPSOffice_Level3)

[1.2.2 国外研究现状](#_Toc25971_WPSOffice_Level3) [2](#_Toc25971_WPSOffice_Level3)

[1.3 本文的研究思路和具体实现方法](#_Toc8407_WPSOffice_Level2) [3](#_Toc8407_WPSOffice_Level2)

[1.3.1 研究思路](#_Toc29638_WPSOffice_Level3) [3](#_Toc29638_WPSOffice_Level3)

[1.3.2 具体实现方法](#_Toc10887_WPSOffice_Level3) [3](#_Toc10887_WPSOffice_Level3)

[1.4 本论文的结构安排](#_Toc25971_WPSOffice_Level2) [3](#_Toc25971_WPSOffice_Level2)

[第二章 数据获取与预处理](#_Toc2868_WPSOffice_Level1) [4](#_Toc2868_WPSOffice_Level1)

[2.1 数据获取及预处理](#_Toc29638_WPSOffice_Level2) [4](#_Toc29638_WPSOffice_Level2)

[2.1.1数据获取](#_Toc16010_WPSOffice_Level3) [4](#_Toc16010_WPSOffice_Level3)

[2.1.2数据预处理](#_Toc351_WPSOffice_Level3) [4](#_Toc351_WPSOffice_Level3)

[2.2数据平稳性检验及处理](#_Toc10887_WPSOffice_Level2) [5](#_Toc10887_WPSOffice_Level2)

[2.2.1数据平稳性检验](#_Toc31192_WPSOffice_Level3) [5](#_Toc31192_WPSOffice_Level3)

[2.2.2非平稳性处理](#_Toc29839_WPSOffice_Level3) [8](#_Toc29839_WPSOffice_Level3)

[2.3本章小结](#_Toc16010_WPSOffice_Level2) [8](#_Toc16010_WPSOffice_Level2)

[第三章 时间序列分析方法介绍与模型训练](#_Toc17397_WPSOffice_Level1) [9](#_Toc17397_WPSOffice_Level1)

[3.1 时间序列分析介绍](#_Toc351_WPSOffice_Level2) [9](#_Toc351_WPSOffice_Level2)

[3.2 常用时间序列分析模型介绍](#_Toc1335_WPSOffice_Level2) [9](#_Toc1335_WPSOffice_Level2)

[3.2.1 自回归移动平均模型ARMA(p,q)介绍](#_Toc4619_WPSOffice_Level3) [9](#_Toc4619_WPSOffice_Level3)

[3.2.2 自回归差分移动平均模型ARIMA(p,d,q)介绍](#_Toc22381_WPSOffice_Level3) [10](#_Toc22381_WPSOffice_Level3)

[3.2.3 长短记忆神经网络LSTM介绍](#_Toc24884_WPSOffice_Level3) [12](#_Toc24884_WPSOffice_Level3)

[3.3 模型训练及构建](#_Toc31192_WPSOffice_Level2) [13](#_Toc31192_WPSOffice_Level2)

[3.3.1ARMA模型识别](#_Toc29512_WPSOffice_Level3) [13](#_Toc29512_WPSOffice_Level3)

3.3.2LSTM模型识别........................................................................................16

[3.4 模型检验](#_Toc11239_WPSOffice_Level2) [16](#_Toc11239_WPSOffice_Level2)

[3.5本章小结](#_Toc29839_WPSOffice_Level2) [18](#_Toc29839_WPSOffice_Level2)

[第四章 服务器搭建与APP开发](#_Toc8407_WPSOffice_Level1) [19](#_Toc8407_WPSOffice_Level1)

[4.1服务器搭建](#_Toc4619_WPSOffice_Level2) [19](#_Toc4619_WPSOffice_Level2)

[4.1.1 WEB服务器与Http协议简介](#_Toc25319_WPSOffice_Level3) [19](#_Toc25319_WPSOffice_Level3)

[4.1.2 Django简介](#_Toc24878_WPSOffice_Level3) [20](#_Toc24878_WPSOffice_Level3)

[4.1.3 服务器搭建的主要实践步骤](#_Toc13928_WPSOffice_Level3) [21](#_Toc13928_WPSOffice_Level3)

[4.2 app开发](#_Toc22381_WPSOffice_Level2) [22](#_Toc22381_WPSOffice_Level2)

[4.3本章小结](#_Toc24884_WPSOffice_Level2) [26](#_Toc24884_WPSOffice_Level2)

[第五章 全文总结与展望](#_Toc25971_WPSOffice_Level1) [28](#_Toc25971_WPSOffice_Level1)

[5.1 全文总结](#_Toc23495_WPSOffice_Level2) [28](#_Toc23495_WPSOffice_Level2)

[5.2 后续工作展望](#_Toc29512_WPSOffice_Level2) [29](#_Toc29512_WPSOffice_Level2)

[致 谢](#_Toc29638_WPSOffice_Level1) [30](#_Toc29638_WPSOffice_Level1)

[参考文献](#_Toc10887_WPSOffice_Level1) [31](#_Toc10887_WPSOffice_Level1)

# 第一章 绪 论

## 1.1 课题研究的背景与意义

## 1.1.1 课题研究的背景

近年来，随着工业的发展，作为人们共享资源的空气正逐渐受到污染,发展中国家城市化以及城镇化的快速发展更是加剧了这一进程。尤其是,在北京和达卡等这样发展中国家的大都市空气污染情况更为严重,即便是空气质量不严重的发达国家,其局部区域也会受到空气污染物的侵害,如美国的芝加哥，法国的巴黎等。空气的污染不仅严重危害了人类的身体健康影响人们呼吸系统，而且降低了空间能见度,给城市交通带来了拥堵或造成潜在交通事故,其中悬浮颗粒物的危害更是巨大,西方国家及美国洛杉矶地区通过统计数据经分析证实了空气污染与地区死亡存在正相关性,并且最终明确雾霾是导致死亡的重要因素之一。我们国家尤其在首都北京及周边的北方地区,雾霾污染已严重影响到了人们的正常生活，雾霾的主要构成是指空气中的细小悬浮颗粒物质,其中,人们常说的PM2.5就是空气中直径小于2.5微米的悬浮物。PM2.5粒径小,富含有大量有毒、有害的物质,在空气中停留久，耗散难，输送远,因此对人们健康和环境的危害非常大,因此在众多污染物中更受人们关注。尤其是近年来随着我国工业发展以及低端产能过剩等原因导致PM2.5污染加剧,人们对空气污染的关注也提到了前所未有的高度。

## 1.1.2 课题研究意义

由于空气污染物与人们的日常生产生活息息相关,更直接关乎空气敏感人群的健康。另一方面,PM2.5的空间分布与城市的建筑、城市结构,城市规划建设的功能用地(公园,工厂,居民区)以及区域环境等因素密切相关,因此其区域性波动较大,有效地监测和预报对于指导人们的生活作息(如是否开窗通风，是否适合户外运动),选择出行时间以及地点,以及对相关治理机构的空气污染的治理有着普遍的指导意义。同时PM2.5的监控、预报和空间流动分析,本身也是构建智慧城市、生态城市的重要一部分。空气污染正逐步侵蚀人类最后的共同的资源，空气污染问题今年来也逐渐被人们所关注，在众多的空气污染成分中，PM2.5对人们的危害最大，能够提前细粒度的预测PM2.5的变化对于人们的出行和作息安排以及决策制定至关重要。

## 1.2 PM2.5的国内外研究历史与现状

## 1.2.1国内研究现状

2011年11月1日开始，我国开始实施环保部发布的《环境空气PM10和PM2.5的测定重量法》。首次对空气污染物PM2.5的测定进行了系统的测量以及预测规范。

2012年2月29日，国务院主持召开国务院常务会议，同意并发布新修订的《环境空气质量标准》，此次修改增加了细颗粒物（PM2.5）浓度限值监测指标。根据安排，2012年在京津冀、长三角、珠三角等重点地区以及直辖市和省会城市开展细颗粒物PM2.5项目监测

目前，我国监测科研技术水平和科学仪器完全能对PM2.5进行监测并预报。早在上世纪90年代，中国就已经在一些发达城市对PM2.5进行监测。数据显示，当时根据在广州、成都、兰州、武汉的几个城市监控点位的监测结果显示，四个城市的PM2.5年均值为50微克/立方米~150微克/立方米，是美国标准值的3倍~10倍。中国环境监测总部在给国家环保部征求意见稿中的回复也提出，近年监测数据表明，我国发达城市中PM2.5已占PM10的60%左右，很有必要增加PM2.5的标准，另外在监测技术方面，我国已基本达到推广PM2.5自动监测技术的技术条件。相对于美国和WHO的推荐值，我国的细颗粒物污染非常严重。沿海地区如青岛、上海、山东、深圳、珠海以及厦门等城市PM2.5的污染水平较低而内陆地区细颗粒物污染严重，这可能与特殊的地理位置和气象条件有关。

另外，由于各城市对PM2.5监测时间不一致，采用的监测方法也不统一，不同的监测方法测得的结果存在系统误差以及数据波动，大大降低了各城市之间PM2.5质量浓度的可比性，因此我国在各大城市开展大规模系统的PM2.5的监测研究路上还有很长路程要走。

## 1.2.2 国外研究现状

在过去的十多年间，美国和一些西方国家开展了大规模对PM2.5研究，主要涉及PM2.5质量浓度的排放清单、时空分布、排放特征、波动趋势以及PM2.5对人体健康影响和大气能见度等方面。美国首都地区是进行细微颗粒物研究最多的地方，研究表明：PM2.5的浓度取决于当地污染物排放、化学变化以及天气条件、地区条件等复杂因素的相互作用；在某些地区，PM2.5及其二次粒子的前体物的浓度主要受地区性和区域性的污染物排放与天气条件的影响；城区单元素颗粒物（元素碳、有机碳）的浓度要高于周边农村地区，而二次颗粒物在地区性的空间分布上则相对均匀受地区影响较小；PM2.5的浓度具有周期性的时间变化且随季节不同而有所不同，不同地区的结果可能也有所不同；

## [1.3 本文的研究思路和具体实现方法](#_Toc466640619)

## 1.3.1 研究思路

本课题的研究核心在于对PM2.5数据的建立合适的模型，并在此模型基础上对未来短期时间的PM2.5值进行预测，本课题还有俩个研究的关键点在于一，开发移动APP来实现预测的操作二，搭建服务器来运载模型，所以本课题的研究思路为：

* 首先采用合适的机器学习相关算法来训练、建立模型，
* 利用相关工具和合适的编程语言开发移动APP
* 搭建个人服务器
* 把模型放服务器上去运作
* 让移动APP和服务器之间能够正常通信

## 1.3.2 具体实现方法

针对上述的5个研究思路的具体实现方法为：

（1）模型的建立

下载相关数据并对数据的预处理，处理成以每天为间隔的时间序列数据，然后采用时间序列数据分析预测方法中的ARIMA算法以及LSTM模型对PM2.5进行建模并预测

（2）APP开发

APP开发采用android studio开发,然后把开发好的APP打包到手机上安装运行。

（3）后台服务器搭建及服务器运行模型

后台服务器采用python3.6+Pycharm+Django搭建一个本地服务器，把建立好的模型放到服务器上，等待APP的数据请求并返回应答数据。

（4）APP和后台服务器的交互

APP和后台服务器的交互主要原理是在APP里面访问服务器地址，APP页面主体有俩个按钮和一个输入框，俩个按钮分别向后台服务器传递俩个不同的固定值，输入框为获取用户自己想要输入的值并传递给服务器，服务器做相应处理并把处理值传递给APP，然后APP接受返回结果并在界面展示。

## 1.4 本论文的结构安排

本文的章节结构安排如下：第一章主要介绍课题研究内容和相关背景、意义以及研究思路和具体实现方法，第二章主要是对本项目中数据获取与处理相关方面的介绍，第三章主要对ARIMA以及其他相关模型进行详细介绍以及模型的建立还有模型之间的对比第四章主要是，APP开发，服务器搭建，客户端与服务器通信等内容的介绍，第五章主要是对本课题研究的总结以及致谢，文末是本论文的参考文献。

**第二章 数据获取与预处理**

数据准备是机器学习的关键步骤之一，数据准备阶段就是从原始数据集到形成最终数据集（输出给模型工具的数据）的所有操作。主要任务是对数据清洗和转换，处理出适合模型工具的数据格式，[时间序列](https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97" \t "https://baike.baidu.com/item/ARIMA%E6%A8%A1%E5%9E%8B/_blank)数据是将某种统计指标的数值，以固定的时间间隔按先后顺序排到以时间单位为间隔的[数列](https://baike.baidu.com/item/%E6%95%B0%E5%88%97" \t "https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E9%A2%84%E6%B5%8B%E6%B3%95/_blank)。

## 2.1 数据获取及预处理

## 2.1.1数据获取

根据本项目以及所用模型对数据要求，在各大数据集网站进行综合比较后最终确定本项目的数据下载来源为http://beijingair.sinaapp.com，其中下载的数据是全国367个城市天气数据 （CSV格式），单项数据里面包括检测日期,监测点名称,城市,经度,纬度PM2.5浓度，PM10、AQI、SO2、NO2、O3、CO、CO2等各个指标的值，一共包括24个字段，然后选取2018年下半年的数据作为本项目的主要研究数据。

2.1.2数据预处理

由于原始数据包含的各项指标较多，处理起来比较复杂所以对于此原始数据的处理方法为主要使用Python的第三方库Pandas、Numpy把数据转换为Dataframe结合数组操作进行处理，首先第一步需要把多余的列给删除，把单个Csv文件数据处理为只含有日期和PM2.5值，这里主要是提取出所需要的列组成新的Dataframe还有对Dataframe.loc（）函数的合理有效运用，然后把多个csv文件数据合并到一个csv文件里面，合并后的数据可能存在缺失值，对于缺失值的处理为在其他各大天气网站查找缺失的值并把它填充进去,这样就形成了完整的数据集，但是数据集中可能还有含有异常值，例如最近7天都是一个比较平稳的一个值但是第8天的值直接飙升或降低到一个异常的数值，对于这种数据的处理为手动改值，把它该为与前后数据相符的数值，经过一系列对数据的预处理后最后得到的数据样例如表3.1所示

|  |  |
| --- | --- |
| Date | PM2.5 |
| 2018/1/1 | 94 |
| 2018/1/2 | 96 |
| 2018/1/3 | 38 |
| 2018/1/4 | 22 |
| 2018/1/5 | 31 |
| 2018/1/6 | 26 |
| 2018/1/7 | 35 |
| ... | ... |

表 3.1

## 2.2数据平稳性检验及处理

## 2.2.1数据平稳性检验

通过2.1的数据获取和处理后，此时获得了以天数为时间间隔的时间序列数据，完成数据准备这一步骤，由于模型的特殊性和对数据的要求，接下来是对数据平稳性进行检验，数据平稳性是比较数据离散性和数据波动性的一个重要指标，如果一个时间序列数据是稳定的那么说明这个数据的波动性和离散性都很小。关于检验时间序列数据是否平稳的方法主要有：数据图观察法，单位根（ADF）检验法,本论文项目在实际实现过程中采用了俩者结合的方法来判断数据平稳性

1. 数据图检验法

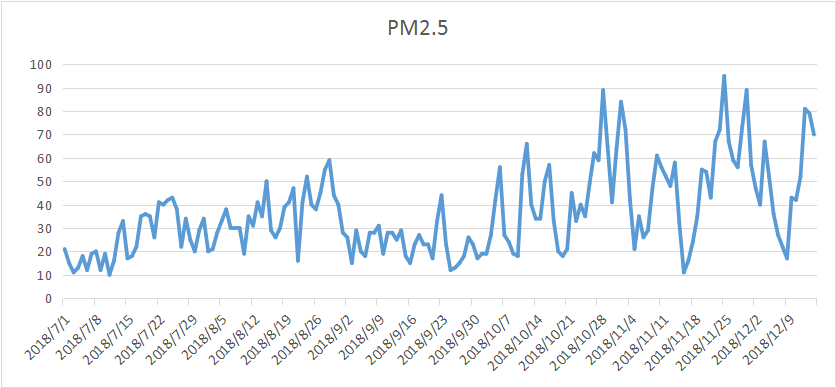
数据图检验法又称观察法，通俗的说就是通过观察给出的一个随机时间序列数据，观察该数据序列随时间的走势图来大致判断它是否是平稳的观察序列的趋势图是否随着时间的变化呈现出某种规律。一个平稳的时间序列在数据图上应该是表现出一个围绕在某个均值范围内不断波动的过程；而非平稳序列则往往表现出在不同的时间段具有不同的均值具有明显的差异性，数据读取和画图采用python进行处理，使用了其第三方库matplotlib,核心代码如下：

data = pd.read\_csv(filepath)

plt.plot(data["date"],data["PM2.5"])

plt.show()

此时得到的数据折线图如图3.2所示：

 图3.2

由图可知，该序列数据随着一个水平上下波动，且起伏水平不是很大，所以初步判断该序列数是平稳的

1. ADF检验

序列数据存在单位根表示，离当前时间很久远的时刻的一个随机冲击对现在的影响仍然没有衰减，如果时间序列存在这种情况，对时间序列的未来值的预测就难以进行，单位根检验法就是指在一定置信水平下，检验序列中是否存在单位根。这是一种常用的单位根检验方法，它的原假设为序列具有单位根，即非平稳，因为存在单位根就是非平稳时间序列，对于一个平稳的时序数据，就需要在给定的置信水平上显著，拒绝原假设。从而证明序列是平稳的

(3)用python进行单位根检验

本项目使用了在python中对时间序列建模通常使用的statsmodel库，因为早期文档实在匮乏，该库在科学计算库和机器学习库排名中长期处于垫底状态，所以目前相关的学习资料也较少，在statsmodels.tsa.stattools.adfuller中可进行adf校验，一般传入一个1维的数据即可，包括list， numpy.array 和 pandas.series等都可以作为输入，其使用过程中默认采用AIC准则，函数的输出为一个tulple其中主要包含5项：

|  |  |
| --- | --- |
| test statistic | 简称为T值，表示t统计量 |
| p-value | 简称为p值，表示t统计量对应的概率值 |
| lags used | 表示延迟 |
| number of observations used | 表示测试的次数 |
| critical value (1,5,10)% | 表示如果t值小于-3.431921，则原假设发生的概率小于1%。其他数值以此类推 |

表3.2 1

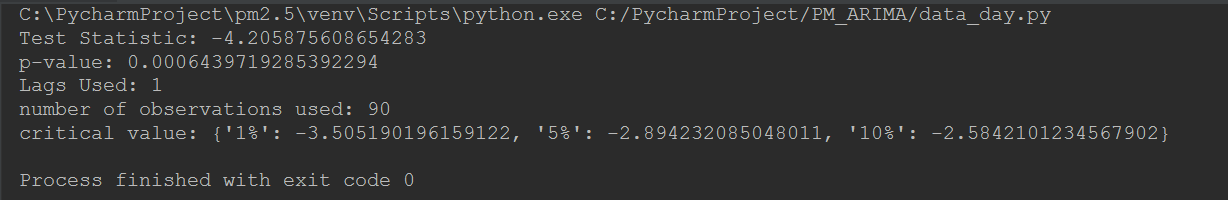
其中t值和p值是最重要的，p值是越小越好，p值要求小于给定的显著水平，p值要小于0.05，等于0是最好的。t值，ADF值要小于t值。其中各水平值的设定意义，1%：严格拒绝原假设；5%：拒绝原假设，10%类推。如果ADF小于1%level值，说明严格拒绝原假设，以此类推。因此如果得到的统计量显著小于3个置信度（1%，5%，10%）的临界统计值时，说明是拒绝原假设的。

核心代码：

data =pd.read\_csv(filepath)

result = adfuller(data["PM2.5"])

输出结果如图3.3所示：

图3.3

本例子ADF值为-4.205，都小于三个level的统计值，即严格拒绝原假设，证明原序列没有单位根，且p-value值为0.00064非常接近零，所以可以得出此序列是平稳的，不用进行平稳性处理。

## 2.2.2非平稳性处理

如果时间序列不稳定，可以通过一些平稳性处理操作去处理数据使得时间序列稳定（比如取对数，差分），然后进行 ARIMA 模型预测，得到稳定的时间序列的预测结果，然后对预测结果进行之前使序列稳定的操作的逆操作（取指数，差分的逆操作），就可以得到原始数据的预测结果。

（1）对数变化

对数变换主要是为了减小数据的振动幅度，减小波动使其线性规律更加明显因为时间序列模型的数据大部分都是线性的，为了尽可能降低非线性的因素，需要对其进行预处理操作。对数变换相当于增加了一个处罚机制，数据越大其惩罚越大，数据越小惩罚越小。需要强调一下，变换的序列需要满足大于零，小于零的数据不存在对数变换

（2）差分

差分就是[离散函数](https://baike.baidu.com/item/%E7%A6%BB%E6%95%A3%E5%87%BD%E6%95%B0/10626142" \t "https://baike.baidu.com/item/%E4%B8%80%E9%98%B6%E5%B7%AE%E5%88%86/_blank)中连续相邻两项之差，在实际中，常常会遇到输入的时间序列经检验是非平稳的，这样就不能采用一些对数据平稳性有要求的模型，例如ARMA模型，对于此类数据通常的处理方法是采用差分的方法将它们变换为平稳的。经差分后，如果时间序列检验为平稳，就对差分后的时间序列进行处理，当一个非平稳时间序列接受了d次差分处理并成为平稳序列后，就能够用一个平稳的ARMA(p,q)模型当作其对应的模型，则称该原始时间序列是一个自回归积分滑动平均时间序列，表示成ARIMA(p,d,q)，其中p为自回归阶数，q为移动平均阶数，d为时间成为平稳时所做的差分次数

## 2.3本章小结

在本章节中主要介绍了在进行项目过程中数据获取与处理方面相关的知识，从众多数据源中进行比较最终确定了数据源，下载数据后对数据进行各种预处理，包括格式处理，异常值处理，缺失值处理等。得到符合模型输入格式的数据后进行了数据的平稳性校验，介绍了数据平稳性检验的相关方法以及非平稳性的处理方式，并对所得数据进行了检验，最终确定了所得数据集为平稳时间序列数据，不用进行非平稳性处理。

**第三章 时间序列分析方法介绍与模型训练**

## 3.1 时间序列分析介绍

时间序列分析方法是[定量线性预测](https://baike.baidu.com/item/%E5%AE%9A%E9%87%8F%E9%A2%84%E6%B5%8B" \t "https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E5%88%86%E6%9E%90/_blank)方法之一。它包括一般统计分析(如自[相关分析](https://baike.baidu.com/item/%E7%9B%B8%E5%85%B3%E5%88%86%E6%9E%90" \t "https://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E5%BA%8F%E5%88%97%E5%88%86%E6%9E%90/_blank)，偏相关分析，谱分析等)，统计模型的建立与训练，以及关于时间序列的最优预测、滤波与控制等内容。一般的经典统计分析都假定数据序列是具有独立性的，而时间序列分析则侧重研究数据序列之间的互相依赖关系。也就是说对离散指标的随机过程的分析统计，所以又可看作是随机过程统计的一种特殊形式。例如，记录了某地区第一个月，第二个月，第三个月…的降雨量，利用时间序列分析的相关方法，就可以对这个地区未来各月的雨量进行预报。

基本思想：根据研究对象在各个时间段的产生值（观察数据），建立能够比较精确地反映序列中所包含的数据之间依存关系的数学模型，并借以对研究对象的未来产生值进行预报。 基本原理：一是承认事物发展的延续性。根据历史数据，就能推测未来事物的发展趋势。二是考虑到事物发展的随机性。任何事物发展都可能受偶然因素影响，为此要利用统计分析中加权平均法对历史数据进行处理。

特点：简单易行，便于掌握，但准确性差，一般只适用于短期预测。

## 3.2 常用时间序列分析模型介绍

常用的时间序列模型有四种：自回归模型 AR(p)、移动平均模型 MA(q)、自回归移动平均模型 ARMA(p,q)、自回归差分移动平均模型 ARIMA(p,d,q),以及神经网络中的LSTM模型, 其中自回归模型 AR(p)、移动平均模型 MA(q)、是自回归移动平均模型 ARMA(p,q)的基础，ARMA(p,q)又是ARIMA(p,d,q)模型的基础特殊形式，只是多了一个非平稳性处理的差分操作，本章主要介绍ARMA模型，ARIMA模型以及神经网络中的LSTM模型。

## 3.2.1 自回归移动平均模型ARMA(p,q)介绍

由模型名字可知，自回归移动平均模型是与自回归和移动平均模型两部分组成。所以可以表示为ARMA(p, q)。p是自回归阶数，q是移动平均阶数。

（一）模型形式



式中符号：p和q是模型的自回归阶数和移动平均阶数

a和是不为零的待定系数，为独立的误差项

（二）模型含义

ARMA模型型结合了AR,MA两个模型的特点，其中，AR可以解决当前数据与后期数据之间的关系，MA则可以解决随机变动也就是噪声的问题。

（三）模型识别

模型的识别方法一般也有两种，一种是自相关函数（ACF），另一种是偏自相关函数（PACF）。这两种方法是识别ARMA模型最有效的方法。可以采用两种函数的拖尾截尾性质来判断该模型的类型。判断方法如表2.2所示：

|  |  |  |
| --- | --- | --- |
| 模型 | ACF | PACF |
| AR(p) | 衰减趋于零 | p阶后结尾 |
| MA(q) | q阶后结尾 | 衰减趋于零 |
| ARMA(p,q) | q阶后衰减趋于零 | p阶后衰减趋于零 |

表2.2

使用自相关函数和偏自相关函数的截尾来判断模型为ARMA模型时，只是得到了p和q的大致取值并不能确定p和q的阶数，为了比较精确的确定p和q的阶数，就必须与常用的定阶准则联合起来应用。如今应用最为广泛的是AIC（最小信息量准则（A-Information Criterion））和BIC准则

（四）ARMA建模步骤

1. 对输入的数据进行判断，判断其是否为平稳非纯随机序列，若平稳则直接进入步骤b；若不平稳则进行数据平稳性处理，处理后才能进入步骤b。
2. 通过自相关和偏自相关函数，并结合AIC或BIC准则对建立的模型进行模型识别和定阶。
3. 完成模型识别和定阶后，进入模型的参数估计阶段。
4. 完成参数估计后，进行模型识别和检验，选择最优模型。
5. 最后，利用准确度高的模型来预测序列的未来变化趋势。

## 3.2.2 自回归差分移动平均模型ARIMA(p,d,q)介绍

ARIMA模型，是指将非平稳时间序列转化为平稳时间序列，然后将平稳的时间序列建立ARMA模型。ARIMA模型根据原序列是否平稳是模型的最大不同点，因此时间序列的平稳性是建模的重要前提。

（一）时间序列的平稳性

时间序列的平稳性的定义：假定某个时间序列由某一随机过程生成，即假定时间序列{Xt}（t=1, 2, …）的每一个数值都是从一个概率分布中随机得到的。如果经由该随机过程所生成的时间序列满足下列条件：

均值E(Xt)=m是与时间t 无关的常数；

方差Var(Xt)=s^2是与时间t 无关的常数；

协方差Cov(Xt,Xt+k)=gk 是只与时期间隔k有关与时间t无关的常数；

如果一个时间序列数据满足上述条件则称经由该随机过程而生成的时间序列是平稳的。该随机过程便是一个平稳的随机过程，平稳时间序列粗略地讲，就是一个时间序列，如果均值没有系统的变化（无趋势）、方差没有系统变化，且严格消除了周期性变化，就称之是平稳的

（二）检验时间序列模型平稳的方法

检验时间序列模型平稳的方法有数据图观察法，单位根检验法一般采用 ADF 单位根检验模型去检验。

（1）图形检验

平稳随机过程的均值和方差函数是常数，所以平稳时间序列的取值必然围绕一个水平值的中心趋势，以相同的发散程度分布波动范围较小。根据这一点，可以从数据分布图形直接对数据是否平稳进行判断。

（2）自相关图检验

原理：平稳时间序列过程的自协方差，和由协方差计算的自相关函数是很小、很快趋向于0，具有截尾或拖尾特征。这些特征正是判断时间序列平稳性的重要依据。因为自相关是相对量指标，方便横向比较和建立一般标准，因此通常利用自相关函数进行判断。

（3）ADF检验，

ADF原假设为，序列存在单位根，即非平稳，对于一个平稳的时序数据，就需要在给定的置信水平上显著，拒绝原假设。 如果序列平稳，则不存在单位根，否则就会存在单位根。

若得到的统计量显著小于3个置信度（1%，5%，10%）的临界统计值时，说明是拒绝原假设的。另外是看P-value是否非常接近0.（4位小数基本即可）

（四）ARIMA建模过程

1. 观测是否为平稳时间序列；对于非平稳时间序列要先进行d阶差分运算，化为平稳时间序列；
2. 经过第一步处理，已经得到平稳时间序列。要对平稳时间序列分别求得其自相关系数ACF 和偏自相关系数PACF，通过对自相关图和偏自相关图的观察和结合AIC的分析，得到最佳的阶层 p 和阶数 q
3. 由以上得到的d、q、p，得到ARIMA模型。然后开始对得到的模型进行模型检验。

同前面的ARMA模型，ARIMA模型也是基于平稳的时间序列的或者差分化后是稳定的，另外前面的几种模型都可以看作ARIMA的某种特殊形式。

## 3.2.3 长短记忆神经网络LSTM介绍

LSTM网络是RNN循环神经网络的一种特殊模式，可以学习长期以来信息，对于所以循环神经网络来说它们都具有神经网络的重复模块链的形式，这些神经网络模块只有一个很简单的结构，LSTM 同样是这样的结构，但是不同于其他单一神经网络层，LSTM重复的模块拥有不同的结构，而且结构之间更加复杂。细胞状态时LSTM的关键，细胞状态类似于传送带，细胞的状态在整个传送带上运行，LSTM主要是来解决避免长期依赖和梯度消失问题，主要思想为门控制和线性连接，门控制保护输入门、输出门、遗忘门主要作用为有选择性的保存和输出历史信息，使得细胞状态可以在传输过程中有效传递，LSTM运作流程图如图3.2所示

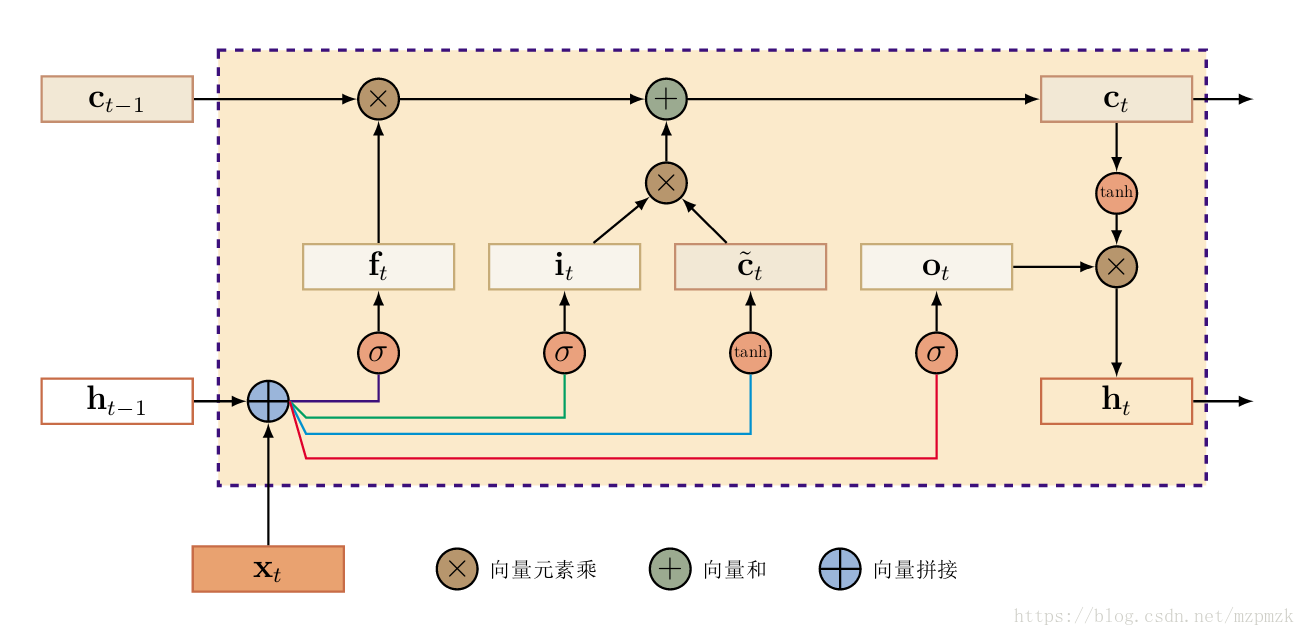


图 3.2

在如上LSTM 时刻t的网络结构中。其中Xt是t时刻的输入，ht−1是t−1时刻隐藏层的输出，Ct−1是t−1时刻历史信息的输出；ft、it和ot分别为 t时刻的遗忘门、输入门和输出门；ct~是t时刻通过变换后的新信息，ct是在t时刻更新过后的历史信息，ht是t时刻隐藏层的输出，由于网上对该神经网络的介绍以及相关学习资料很大，也有了相关的第三方库，所以在本项目中我们也直接是站在巨人的肩膀上看太阳，不用重复造轮子。直接使用现有的第三方库。

## 3.3 模型训练及构建

机器学习方法的核心在于模型的训练和建立，模型效果的好坏将直接影响预测结果，所以本小节将详细介绍本项目在实施过程中关于模型建立的具体步骤，时间序列分析就是通过统计和分析时间序列，根据时间序列所反映出来的某个现象的发展过程、趋势和方向，进行类推或分析，借以分析这个研究指标的数值在下一段时间或以后若干个时间段内可能达到的水平

## 3.3.1ARMA模型识别

由于数据是平稳的所以采用的模型为ARMA模型，模型识别主要采用了观察ACF、PACF图和循环阶数寻找最优阶以及模型自动定阶三种方法相结合

1. 观察ACF、PACF图

绘制ACF、PACF图的核心代码如下：

data = pd.read\_csv(filapath)

xdata = data["PM2.5"]

f = plt.figure(facecolor='white')

ax1 = f.add\_subplot(211)

plot\_acf(xdata, lags=31, ax=ax1)

ax2 = f.add\_subplot(212)

plot\_pacf(xdata, lags=31, ax=ax2)

plt.show()

数据的ACF、PACF图如图3.4所示：

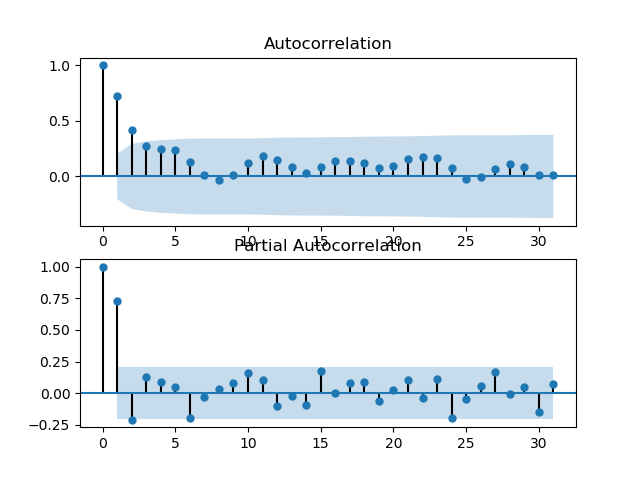


图 3.3.3

图3.4

ARMA（p,q）模型自相关，偏相关图定阶方式为：q阶后衰减趋于零 p阶后衰减趋于零，由上图可知p值可选取的值有1或者2，q可选取值为1，所以模型形式有ARMA(1，1)和ARMA(2，1)俩种，这一步我们经过观察的方法得到了模型的初步形式，想要得到精确的结果或者对模型进行验证还需其他俩种方法

（2）循环阶数寻找最优阶

此方法为定义俩个for循环，逐一尝试不同值的p和q的组合，计算所有组合的BIC值，找出其中最小的组合就是最优解核心代码如下：

bic\_matrix = [] # bic矩阵

for p in range(5):

tmp = []

for q in range(5):

try:

tmp.append(ARMA(xdata, (p, q)).fit().bic)

except:

tmp.append(None)

bic\_matrix.append(tmp)

这时从bic\_matrix中国找出最小值，此方法可以依据BIC准则识别模型的p, q值，通常认为BIC值越小的模型相对更优，最终结果显示Bic最小的组合为p=1,q=1

（3）模型自动定阶

此方法为第三方库statsmodels中自带方法arma\_order\_select\_ic（），该方法输入为一维数据data,最大p和最大q值，以及判断模型优良的识别准则，默认为BIC,核心代码如下：

a = arma\_order\_select\_ic(xdata, max\_ma=5, max\_ar=5)

print(a.bic\_min\_order)

结果输出为（1,1）

此方法与方法2的底层实现原理基本相同，唯一不同点在于可以指定不同的识别准则，由上面三个方法结合比较可得最终模型形式为：ARMA(1，1),至此模型的训练和识别工作完成

3.3.2 LSTM模型识别

本次LSTM模型的建立时基于keras来建立的,由于神经网络的层数并不越多越好，所以本次项目采用了俩层，建立模型的核心代码如下：

model = Sequential()

model.add(LSTM(units=100,input\_shape=(None,1),return\_sequences=True)

model.add(LSTM(units=100)

model.add(Dense(units=1))

model.add(Activation('linear'))

model.compile(loss='mse',optimizer='adam')

当进行模型拟合时，核心代码如下：

## model.fit(X\_train, y\_train, batch\_size=20, epochs=80, validation\_split=0.1)

## 3.4 模型检验

模型检验方法为将预测数据与真实数据做比较，并采用相关误差判别方法进行判别模型效果好坏，图3.4.1是检验ARMA模型的结果，图上俩个曲线是由真实数据和通过上面得到模型进行预测得到的数据进行比较，通过对图的观察可知，检验下来，模型的具有一定的可信度

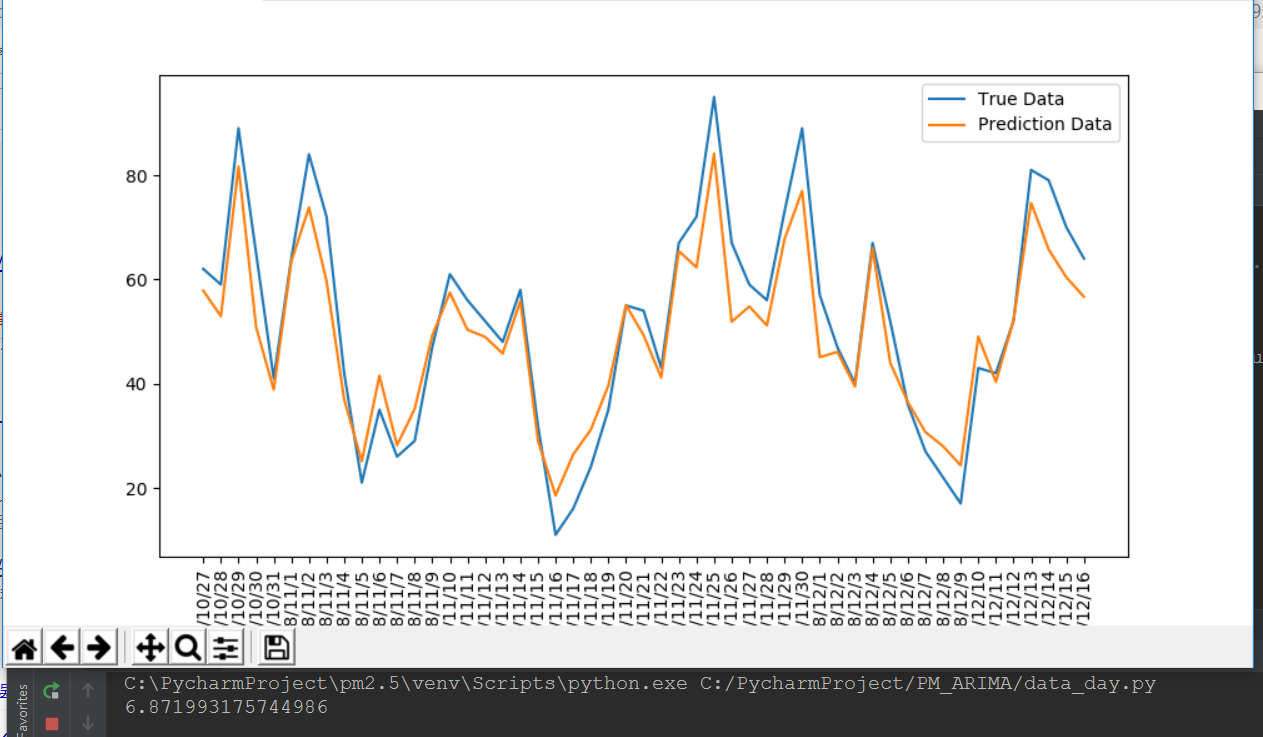


图 3.4.1

图3.4.2是检验LSTM模型的结果，图上俩个曲线也是由真实数据和通过模型进行预测得到的数据进行比较，通过对图的观察可知，检验下来，模型也是具有具有一定的可信度，但对比ARMA模型来说效果较差

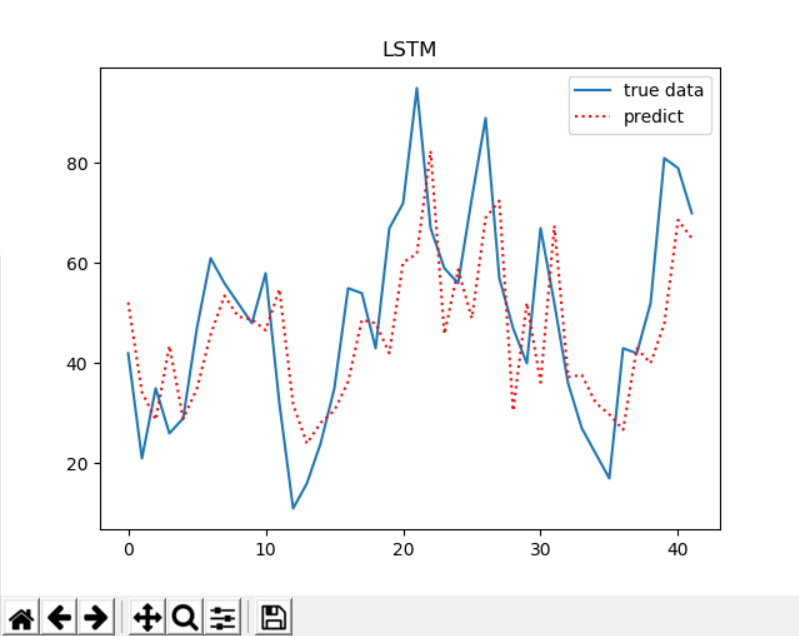


图 3.4.2

我们对模型还要进行误差检测，这里主要采取的检查方法有均方根误差(RMSE)检验，均方根误差检验是观测值与真值偏差的平方和与观测次数m比值的平方根。是用来衡量观测值同真值之间的偏差，还有一个就是平均绝对误差（MAE），平均

绝对误差是绝对误差的平均值能更好地反映预测值误差的实际情况.对ARMA模型预测结果的误差检验结果如图3.4.3所示：



图 3.4.3

其中均方根误差检验结果为：6.8719，平均绝对误差检验结果为：6.739

图3.4.4为LSTM模型的误差检验结果



图 3.4.4

其中均方根误差检验结果为：14.58，平均绝对误差检验结果为：11.94，通过比较俩个模型的预测效果以及结果分析最终我们确定arma模型最为本项目的使用模型，至此模型训练工作完成

3.5本章小结

本章一开始介绍了时间序列数据的定义，然后概括了时间序列数据的分析方法和常用分析模型，随后分别详细的介绍了常用的四种时间序列模型有四种：自回归移动平均模型 ARMA(p,q)、自回归差分移动平均模型 ARIMA(p,d,q),以及神经网络中的LSTM模型，分别从模型含义，模型形式，模型识别，模型工作流程等各方面对各个模型进行了详细的阐述，然后训练了ARMA模型和LSTM模型，并对俩种模型进行了分析比较最终确定了ARMA模型为最优模型

**第四章 服务器搭建与APP开发**

## 4.1服务器搭建

## 服务器搭建也属于本项目的核心关键点之一，作用在于能使3.1建立好的模型放在服务器上运行使之能够被客户端访问，在本项目中采用python+Django来搭建一个简单的本地服务器，所以服务器搭建部分在整个项目中属于比较简单的一部分，实现起来基本没太大难度。

## 4.1.1 WEB服务器与Http协议简介

Web服务器是指驻留于[因特网](https://baike.baidu.com/item/%E5%9B%A0%E7%89%B9%E7%BD%91" \t "https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)上某种计算机的程序。当客户端连到服务器上并发出请求时，服务器将处理该请求并将处理结果发送到该客户端上，附带的信息会告诉客户端其他一些连接的详细信息。服务器主要使用[HTTP](https://baike.baidu.com/item/HTTP" \t "https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)（[超文本](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC" \t "https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)[传输协议](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE" \t "https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)）与客户端浏览器进行信息交流，所以人们常常也把它们称为HTTP服务器。Web服务器本质上来说就是一个Socket服务端，在不停地等待着客户端的请求，然后针对每一个客户端的请求进行处理，处理完毕就即时关闭连接。而我们的浏览器相当于一个Socket客户端，通过TCP协议向服务端发送HTTP请求报文。

HTTP协议是用于从万维网服务器传输超文本到本地浏览器的传送协议，传递数据方式是基于TCP/IP通信协议，http是基于请求响应模式的。客户端向服务器发送一个请求，请求头包含请求的方法、 URI、客户端信息、以及包含请求修饰符、协议版本的消息结果。服务器则以一个状态行作为响应，相应的内容包括响应结果，消息协议的版本、响应编码加上包含服务器信息、实体元信息等HTTP协议同时也是无状态协议，依赖于近乎瞬间的请求处理。请求信息被立即发送，理想的情况是 没有延时的进行发送处理，不过，这只是理想状态，延时还是客观存在的。HTTP有一种内置的机制，在消息的传递时间上由一定的灵活性：超时机制。一个超时就是客户机等待请求消息的返回信息的最长时间

每一个HTTP请求都会经历三个步骤：请求-处理-响应：每当我们在浏览器中输入一个URL时会被封装为一个HTTP请求报文发送到Web服务器，而Web服务器则接收并解析HTTP请求报文，然后针对请求进行处理。最后将要返回的内容转为输出流并封装为HTTP响应报文发送回浏览器。浏览器接收到响应报文后会加载返回内容中的HTML、CSS与JS并显示在页面中，最后成为我们看到的最终效果。

## 4.1.2 Django简介

Django 框架是一个Python定制框架，它源自一个在线新闻 Web 站点，于 2005 年以开源的形式被释放出来。Django 框架的核心组件有：

* 用于创建模型的[对象关系映射](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1%E5%85%B3%E7%B3%BB%E6%98%A0%E5%B0%84" \t "https://baike.baidu.com/item/django/_blank)
* 为最终用户设计的完美管理界面
* 一流的 URL 设计
* 设计者友好的模板语言
* 缓存系统

Django是基于[MVC](https://baike.baidu.com/item/MVC" \t "https://baike.baidu.com/item/django/_blank)构造的框架。在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 的设计模式更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。它们各自的职责如下：

|  |  |
| --- | --- |
| 层次 | 职责 |
| 模型（Model），即数据存取层 | 处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。 |
| 模板(Template)，即表现层 | 处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。 |
| 视图（View），即业务逻辑层 | 存取模型及调取恰当模板的相关逻辑。模型与模板的桥梁。 |

从以上表述可以看出Django 视图不处理用户输入，将MVC中的视图进一步分解为 Django视图 和 Django模板两个部分，分别决定展现哪些数据和如何展现，使得Django的模板可以根据需要随时替换，而不仅仅限制于内置的模板。

至于MVC控制器部分，由Django框架的URL配置来实现。URL配置机制是使用正则表达式匹配URL，然后调用合适的视图处理函数（view）。URL配置对于URL的规则没有任何限制，完全可以设计成任意的URL风格，不管是传统的，还是另类的。框架把控制层给封装了，与数据交互这层都是数据库表的读,写,删除,更新的操作。在写程序的时候，只要调用相应的方法就行了，在完成项目时把控制层东西交给Django自动完成了。只需要编写非常少的代码完成很多的事情。

一个基于Django的http请求流程为：

1. 当访问url的时候，Django会首先加载URL配置。然后按顺序逐个匹配URL配置里的URLpatterns。
2. 如果找到则会调用相关联的视图处理函数，并把HttpRequest对象作为第一个参数
3. 最后该view函数负责返回一个HttpResponse对象

**4.1.3 服务器搭建的主要实践步骤**

（1）首先安装和搭建Django

在命令行输入pip install django 即可完成安装，搭建django服务器框架也仅需一条简单的命令 django-admin startproject projectname,通过上面俩个步骤完成了Django的安装和服务器的搭建

（2）新建App和编写视图函数

这里的App相当于java项目里面的module，进入项目后通过命令python manage.py startapp appname，即可完成App的创建，然后在views.py里面编写视图函数，视图函数核心代码如下：

def arma\_forecast(request,day\_num):

model = ARMA(dta, order=(1, 1)).fit(disp=0, method='css')

a = model.forecast(steps=int(day\_num))

data = [int(i) for i in a[0]]

return HttpResponse(json.dumps(data))

该代码函数名为arma\_forecast(配置url映射时会用到)，函数有俩个输入参数，第一个为 固定的request,每一个视图函数都具有这个参数，第二个参数为day\_num代表意思为预测为了几天的值，函数输出为预测结果

（3）配置url映射

Url的映射在urls.py文件中编写，核心代码如下：

urlpatterns = [

url(r"^$",views.index),

url(r'^admin/', admin.site.urls),

url(r"^forecast/(\d+)",views.arma\_forecast),

]

前面俩个为django自动生成的，后面一个为自己编写的，指定一个url,后面调用对应的视图处理函数，具体意思为当当前请求的url为forecast/int结尾时（例如http://127.0.0.1:8000/forecast/4），django会调用arma\_forecast这个函数，并把这个url结尾的数字传递给该函数。并返回处理结果。

（4）运行服务器

通过命令python manage.py runserver (可指定地址和端口号，默认为127.0.0.1:8000)即可启动服务器，在浏览器输入http://127.0.0.1:8000/

forecast/4时即可得到返回数据，返回数据为数组格式，这里的返回数据最终是要在APP里面展示，所以此次只是返回结果，没有特殊的样式

## 4.2 app开发

## APP开发也属于本项目的核心关键点之一，作用在于访问3.2搭建好的服务器并展示数据，在本项目中采用java+Android Studio来搭建一个简单的PM预测App，因为之前学习的主语言是python，对java了解不多，而且也没有接触APP开发，所以App开发这部分在整个项目中属于最困难的一部分，这部分的开发个人觉得是任务难度最大，花费时间最长的一部分。因为App开发设计的知识点很多，涉及到java的多线程，网络编程，安卓的各大组件，布局，资源管理，数据存储等等，自己也不是特别了解，所以本小节不会对各个知识点进行阐述，只是对实际项目中涉及到的知识点进行详细介绍。主要介绍点有以下几点

一、页面布局

安卓开发的布局方式主要有以下几种FrameLayout（帧布局），LinearLayout（线性布局），RelativeLayout(相对布局)，GridLayout(网格布局)AbsoluteLayout(绝对布局)等，线性布局是程序中最常见的布局方式之一，也是本项目才用的布局方式，所以下面只对它进行详细介绍

线性布局存在水平线性布局和垂直线性布局两种划分形式，具体是那种布局形式是通过android:orientation这个属性来控制的如果这个属性值为："horizontal"那就是水平线性布局如果属性值是"vertical"那就是垂直布局，水平布局和垂直布局的区别简单的来说，就是如果你是垂直线性布局，那么你的视图就是呈垂直的方向依次排下来，如果你的是水平线性布局的话，那么你的视图就会呈水平的方向顺序排下来。在线性布局管理器允许下，每一个子视图可以指定一个weight属性，用来控制每一个子视图在空间内的大小，垂直布局每一行除了自己嵌套的控件之外只能有一个控件；水平布局只有一行，所有的控件依次从左向右排列；

二、基本控件

# 安卓的八大常用控件：TextView，EditText，Button，CheckBox，RadioButton，ImageView，ImageButton，ProgressBa本项目中用到的控件有TextView,EditText，Button这三个最基本的控件，也是本系统所用到的控件，下面将对这三个控件一一进行详细介绍

1. TextView

从字面意思上看，TextView就是文本视图，只是用来显示文字的。要想在Activity中显示文字, 我们需要在相应的布局文件去添加相应的控件标签。这些标签可以确定控件的位置，大小，颜色等属性。下面是在实际项目中Activity显示一个TextView。布局核心代码如下：

<TextView

android:id="@+id/title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:textColor="#EE7942"

android:text="PM2.5预测系统"

/>

android:id属性在所以控件里面都代表着这个控件的Id,也就是这个控件的唯一标识符，在java代码中我们可以通过findViewById()方法来通过Id获取控件。

android:layout\_width属性代表着控件的宽度，该属性的值是wrap\_parent, 表示控件的宽度是动态改变的，根据内容的宽度进行改变。

android:layout\_height属性代表着控件的高度，该属性的值是wrap\_content,

表示控件的高度也是动态改变的，根据内容的高度进行改变。

android:gravity属性代表着这个控件中文字对齐方式，文字对齐有多种方式，我们在此选的是center,居中显示。

android:textSize属性代表着这个控件中文字的型号，也就是文字的大小。

android:textColor属性设置的是这个控件中文字的颜色。

android:text属性就是用来设置这个控件显示的值的。

1. EditText

如果想要为Activity添加一个输入框，在页面布局代码中输入框的类型和标签都是EditText。功能是接收用户输入的数据的。项目中用到的EditText控件布局代码如下：

<EditText

android:id="@+id/edit\_num"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="请输入天数"

/>

### 其中android:hint属性后边是一个字符串，功能是提示用户该输入框是做什么的，用来指定提示语，关于其他各个属性代表意思参照Textview,

(3)Button

在Android中的按钮控件叫Button。我们要使用按钮时需要在Activity对应的布局文件layout中添加一个Button, 项目中具体的布局代码如下所示：

<Button

android:id="@+id/main\_bt2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="预测后天PM"

/>

在采用线性布局的垂直布局方式并添加各个控件之后得到的页面布局如图4.2下所示：



图 4.2

其中“预测明日PM”这个按钮作用为预测明天的PM2.5值，相当于给服务器的arma\_forecast函数传入的天数为1点击后，页面下方会展示结果，同理“预测明后俩天PM”这个按钮作用为预测后俩天的PM2.5值，相当于给服务器的arma\_forecast函数传入的天数为2点击后同样会展示数据，下面的输入框为自定义输入数据，输入后点击确定按钮app会把参数传递到服务器然后接受返回数据并展示

三、app与服务器通信

要进行安卓的网络开发，首先第一步在配置文件AndroidManifest.xml中添加访问网络的权限：<uses-permissionandroid:name="android.permission.INTERNET"/>

然后还有非常重要的一点就是对于APP来说，网络请求、资源处理一定不能在主线程中进行，一定要新开启一个子线程在子线程中进行。因为网络请求一般有几秒钟左右的延时，在主线程中进行网络请求会造成主线程的停顿，主线程应该只进行ＵＩ绘制，页面控制等非延时操作，例如网络请求、等待响应、资源下载等各种耗时操作都应该放到子线程中来进行。

1. App的网络连接

App的网络连接主要是实例化一个URL对象，然后使用了URL的openConnection()方法，这个方面方法会返回一个URLConnection对象，该对象表示应用程序和URL之间的通信连接。程序可以通过URLConnection对象的实例向这个URL发送请求，读取URL引用的资源。通常创建一个和URL的连接，并发送请求、读取此URL引用的资源接受返回数据需要如下几个步骤：

①通过调用URL对象的openConnection()方法来创建URLConnection对象；

②设置URLConnection的参数和普通请求属性；

③如果只是发送GET方式的请求，那么使用connect方法建立和远程资源之间的实际连接即可；如果需要发送POST方式的请求，则需要获取URLConnection实例对应的输出流来发送请求参数；

④远程资源变为可用，程序可以访问远程资源的头字段，或通过流来处理返回的资源数据。

（2）子线程开启

项目中开启子线程的方法为实现Runnable接口并重新run方法，部分具体代码如下：

public class My\_thread implements Runnable{

public void run() {

//具体网络连接加数据处理代码

}

然后在run方法中,写了关于网络连接以及返回数据处理的，主要为流的处理方式，利用InputStreamReader将字节流输出转换为字符流，核心代码如下：

{

String url="http://192.168.31.235:8000/forecast/"

URL urlobj = new URL(url);

URLConnection connobj = urlobj.openConnection();

BufferedReader in = new BufferedReader(new InputStreamReader( connobj.getInputStream()))

Message msg = Message.obtain();

myhandler.sendMessage(msg);

}

此处的Url地址为服务器的具体地址，所以在启动服务器时不能使用默认地址，要指定启动地址和端口号，所以启动服务器时要使用python manage.py runserver 192.168.31.235:8000这种方法来启动

四、结果展示

结果展示首先是通过获取android:id来获取布局中的TextView也就是要展示的位置，然后自定义一个消息处理器My\_handle来处理子线程中网络请求的结果，My\_handler继承Handler类，然后重写其中的handleMessage方法，在handleMessage方法中对服务器返回结果进行处理，并通过TextView.setText()的方法把结果进行展示。

**4.3本章小结**

本章节属于本论文的核心章节，总共分为二个小节分别对项目中的服务器介绍，服务器搭建，APP基本控件介绍还有APP开发，这俩个最重要的关键开发点进行详细介绍。每小节中又分别对各个步骤以及具体实现方法进行了详细阐述，包括了服务器的搭建步骤，网络连接，接受返回数据并进行返回数据展示等方面进行详细介绍。从Django的原理、下载、基础命令以及基本设计模式等对服务器搭建进行介绍。至于App开发方面，由于Java和安卓开发的生态体系太过庞大，所以只是对项目中设计到的点主要包括页面布局，基本控件使用，网络请求等几个方面进行细述。

第五章 全文总结与展望

## 5.1 全文总结

本文以时间序列预测方法为研究背景，主要对相关的时间序列分析算法进行了深入研究。同时还对服务器搭建和App开发进行了一些探索研究，在研究时间序列分析算法的过程中，主要研究了其中的自回归移动平均模型ARMA(p,q)、自回归差分移动平均模型ARIMA(p,d,q)，以及神经网络中的LSTM模型。在介绍几个模型的时候分别从模型含义、模型形式、模型使用场景、模型运作流程、模型识别等这几个方面来对这几个时间序列分析算法模型进行详细探讨。同时根据课题要求和课题数据样例还有各类模型的使用特点以及模型之间的比较，最终采取自回归移动平均模型ARMA(p,q)作为本课题的重点研究与实现对象。在选取适用的模型后通过图形定参、手动定参以及模型的自动定参等方式相结合最终确定了模型的最终样式。

同时在服务器搭建方面采取了python+Django来搭建本次项目的服务器，Django是一种开源的大而且全的Web应用框架，是由python语言来编写的。他的设计模式也是采用了MVC模式，同时具有强大的数据库功能、自带强大的后台功能、具有模板系统、开发难度小等几个优点，所以Django框架作为本次项目所采取的服务器框架，通过把训练好的模型放到服务器上运作，并通过URL配置传参来访问模型的方式做好服务器响应的准备，同时服务器开发方面也属于本项目较为简单的一部分。

在APP开发方面，采用了java+Android Studio来开发本次项目的APP，正如前文所言，此项目的APP界面较为简单，它的功能仅包含传参和页面展示，所以APP界面包括俩个按钮和一个输入框以及一个展示界面。俩个按钮和输入框，其功能都为向后台服务器传参，俩个按钮的参数为固定的，输入框的参数是由用户输入决定，展示界面用于接受服务器返回数据并进行展示。

在本课题中，总体来说第一部分模型构建花费时间最长占总工作量的60%，其中包括数据获取、数据处理、模型选择、模型训练、模型识别等，处理好模型之后就是服务器开发，此部分开发较为简单工作量占比10%左右，然后是APP开发，在进行此部分的开发时，由于之前没有相关经验，在环境搭建和工具选取等前期准备上花费时间较长，在学习APP基础知识和实际开发中也是花费了较多时间，最后做出了一个简易的APP，最后在进行开发客户端与服务器交互的过程中，遇到的困难也比较多，涉及到网络资源访问，参数传递，接受数据等知识，尽管开发难度较大，但最终还是完成了项目

## 5.2 后续工作展望

## 近年来，时间序列分析方法的研究得到了迅速发展，时间序列分析覆含的领域很广，相关知识点也很多，本论文的研究点对于这个领域来说只是冰山一角，所以结合本文研究工作和内容，个人觉得本项目仍有以下方向值得进一步研究了

（一）改进APP界面

在本项目中由于涉及的APP开发的知识较少，只是具有一些基础控件的功能，加上对APP开发知识掌握程度的欠缺，所以做出来的APP较为简陋，后期可以在不影响使用结果的基础上改良并美化APP

（二）实时预测

目前项目遇到可以改进的点是做到实时预测，目前此项目预测的基础来自于已准备的数据集，例如准备的训练数据是到2月15号，此时在app里点击“预测明日PM”这个按钮时，他返回的预测数据是2月16号的，并不是真正意义上的明天，并不能做到实时预测。目前的拟解决办法有俩个，一是在服务器里增加一个爬虫功能，当点击预测按钮时，先触发爬虫功能去天气网站爬取截止当日的历史PM数据，然后用此数据训练模型再返回结果从而达到实时预测。此方法实现难度略大、服务器响应时间长。要涉及爬取数据，再训练，然后才预测返回结果。二、定期更新数据集，从而更新服务器里面的模型，此方法也是方法也是方法一的特殊情况，实现难度较简单，但在预测时可能会存在误差。

致 谢

本论文的研究工作是在我的导师付波老师耐心指导下完成的，在完成论文设计的过程中，从开题报告开始付老师就细心指引和教导，共同商量毕业论文设计的出发点，研究思路，研究步骤，项目的难点以及注意点。在后期的研究中在项目进行遇到困难向老师请教时也得到了耐心的解答或者解决思路，做到有问必答。在论文撰写的过程中遇到困难老师也即使对我遇到的问题和疑惑给与耐心教导，提出了很多有效的改良性意见，投入了很多的心血和精力。

同时还要感谢我的同班同学白智力，因为在毕设过程中有段时间我是外出实习，在办理毕业设计的相关手续，比如文件打印，相关文件签字时，都是该同学多次跑去付老师办公的地方找付老师给我签的。还有我的室友，中期答辩按照学校手续办理请假之后，室友也是积极主动的帮我代述我的毕业设计中期报告

最后感谢学院的老师，辅导员等，在毕业设计过程中给与了我们很大的支持与帮助，能够及时通知我们相关信息并且督促我们完成毕设，在询问一些其他信息时也总能在第一时间给我们答复。

参考文献

1. Glodxwang 时间序列预测全攻略 <http://www.36dsj.com/archives/44065> 2017.6.16
2. 张焕明 时间序列分析 安徽财经大学 [http://mooc.chaoxing.com/course/](http://mooc.chaoxing.com/course/3037871.html)

[3037871.html](http://mooc.chaoxing.com/course/3037871.html) 2016.5

1. Man\_world LSTM原理与简介https://blog.csdn.net/mzpmzk/article/

detail/80548927 2018.6.2

[4]李瑞丽，沈薇，利用统计学方法建立消费信贷评分模型].经济与管理，2005

[5]刘明，经济时间序列的ARIMA类模型构建].统计与决策，2014

[6]杨颖梅.基于ARIMA模型的北京居民消费价格指数预测D].统计与决策，2015

[7]陆波，闪思韬，闵红星，岳晓玲，郭忠琴。应用ARIMA模型预测麻疹发病率的可行性研究J].中国卫生统计，2015

[8]吉乔伟，毛根海，郑冠军，等.基于改进ARIMA模型的时用水量预测U].江南大学学报（自然科学版），2008

[9]王黎明, 王连, 杨楠. 应用时间序列分析. 复旦大学出版社, 2009  
[10]王燕. 应用时间序列分析. 中国人民大学出版社, 2005  
[11]王振龙. 时间序列分析. 中国统计出版社, 2000  
[12]王振龙, 胡永宏. 应用时间序列分析. 科学出版社, 2007  
[13]何书元. 应用时间序列分析. 北京大学出版社, 2004  
[14]安鸿志. 非线性时间序列分析. 上海科学技术出版社, 1998  
[15]恩德斯, 谢志超. 应用计量经济学: 时间序列分析. 高等教育出版社, 2006  
[16]沃尔特, 恩德斯. 应用计量经济学时间序列分析. 高等教育出版社, 2006  
[17]吴喜之. 应用时间序列分析（R软件陪同）. 机械工业出版社, 2014