Computer Science and Application 计算机科学与应用, 2023, 13(1), 1-13
Published Online January 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.131001

Hans汉斯

新冠肺炎疫情预测及可视化系统的设计与实现

张宽宽，张翠平，陈菁菁，王子豪

北京信息科技大学计算机学院，北京

收稿日期：2022年12月2日；录用日期：2022年12月30日；发布日期：2023年1月9日

摘　要

新型冠状病毒所引起的肺炎疫情至今仍余孽未消，世界人民的生命健康和安全仍处于无形的威胁下。本论文介绍了如何利用Python网页数据爬取和Pyecharts可视化技术实现对中国和全球疫情的数据可视化，并利用神经网络技术对中国疫情的发展趋势进行大致预测，使群众能够清楚地了解疫情信息并做好自身防护工作，以便早日消除疫情隐患。

关键词

新冠肺炎，数据爬取，可视化，神经网络

Design and Implementation of COVID-19 Prediction and Visualization Systems

Kuankuan Zhang, Cuiping Zhang, Qingqing Chen, Zihao Wang

School of Computer Science, Beijing Information Science & Technology University, Beijing

Received: Dec. 2nd, 2022; accepted: Dec. 30th, 2022; published: Jan. 9th, 2023

Abstract

The pneumonia epidemic caused by the new coronavirus is still unencumbered, and the life, health and safety of the people of the world are still under invisible threats. This paper describes how to use Python web data crawling and Pyecharts visualization technology to visualize the data of the epidemic situation in China and the world, and use neural network technology to make a rough prediction of the development trend of the epidemic in China, so that the masses can clearly understand the epidemic information and do a good job in self-protection, so as to eliminate the hidden dangers of the epidemic as soon as possible.

文章引用: 张宽宽, 张翠平, 陈菁菁, 王子豪. 新冠肺炎疫情预测及可视化系统的设计与实现[J]. 计算机科学与应用, 2023, 13(1): 1-13. DOI: 10.12677/csa.2023.131001

Keywords

COVID-19, Data Crawling, Visualization, Neural Networks

1. 新冠疫情可视化及预测研究的发展现状

研究背景和国内外发展现状

截至到目前，新冠疫情新增确诊人数在国内起伏较小，社会安定，但把视野放到全世界上，其他国家还在水深火热之中。在大数据技术发展快速的今天，海量的数据在网络上不断生成，只有正确利用这些现有数据才能让民众更好了解新冠疫情[1] [2]。国内针对疫情大数据平台的研究早在 2020 年初，新冠肺炎在全国大规模爆发之际便应运而生。以江苏省为例，江苏省测绘工程院等单位以官方发布的新冠肺炎疫情相关数据为基础，综合运用时空大数据分析和跨平台可视化等技术，设计并实现了基于微服务架构的江苏省疫情大数据平台[3]。此外，国外对疫情大数据可视化平台的架构与实现也在同时进行着。以欧洲为例，将 9 月 14 日欧洲的新冠肺炎疫情状况信息数据导入到 Hive 与大数据架 Hadoop 中进行数据分析，借助 Tableau 组件(数据可视化软件)将其可视化，展现出 9 月 14 日的新冠疫情状况，导出了不同指标的一系列图表[4]。

在疫情预测方面，在新型冠状病毒爆发开始，全世界的学者就着手进行确诊人数分析和增长趋势预测，发表数目较多的学术论文。王旭艳团队采用时间序列分析，使用求和自回归移动平均模型(ARIMA)对新冠疫情数据进行重新建模并预测，结果与现实拟合度高[5]。盛华雄等人将疫情传播分为两个阶段，即控制阶段采用差分递推方法和差分递推方法分析预测，自由传播阶段运用 Logistic 模型比较分析，论证了提早采用防疫措施的好处[6]。而曹盛力等人的团队通过修正的 SEIR 模型对湖北省疫情进行评估和预测，结果表明了修正的 SEIR 传染病动力学模型可用于对 COVID-19 传播态势的分析预测[7]。任晓龙等人采用 Logistic 回归方法对美国新冠肺炎确诊人数建立预测模型，通过实际数值进行拟合预测分析，结果发现在新冠肺炎疫情发作的前中期，Logistic 模型拟合预测的情况与实际数据基本吻合，具有较高的预测精度，可用于新冠肺炎感染确诊人数的初期预测[8]。基于这样的研究背景，以及考虑到神经网络技术可以对通过机器学习对时间序列曲线进行近似拟合，采用神经网络技术来实现对肺炎疫情趋势的大致预测。

2. 新冠肺炎疫情预测及可视化系统的研究内容

2.1. 对包含国内外疫情信息的网页进行数据爬取和数据清洗

疫情信息蕴藏在国家卫健委所统计的数据里，这些数据包括国内数据和海外数据，国内数据包含了全国各省市的每日新增确诊数，累计确诊数，新增无症状感染者病例数，死亡病例数，治愈病例数，高风险，中风险，低风险地区数，而国外数据包含国外各个国家的新增病例数，新增死亡数等信息，但数据内容并不全面，这是因为一些地区或一些国家的疫情数据信息并没有对外公布。每日腾讯疫情，丁香园以及网易新闻等第三方网页都会对这些以及公布了的疫情数据进行展示，对这些数据源网页进行爬虫是实现疫情数据可视化必要的准备工作。此外，还要对获取的数据进行预处理，保证数据质量的关键步

骤是数据清洗。

最后，在进行数据可视化工作前，为了方便进行数据管理，提高数据重复利用时程序的效率，应将收集和清洗后得到的数据存入数据库中。

2.2. 疫情数据处理分析和可视化

数据可视化是借助图形化的方法，清晰有效的将数据中所蕴涵的信息展示出来。常见的数据可视化图形比如柱形图，饼图，折线图等能够直观地呈现出数据的特征以及历史数据的趋势。在本项目中，我们借助中国和全球的疫情地图，根据各地区确诊数等数据的大小对区域地图版图进行不同程度的染色，疫情确诊和死亡人数越多的地区所对应的地图模块的颜色就越深，这样可以使得疫情的基本状况能够一目了然。除此之外，一些疫情信息的细节可以通过交互式图表进行进一步的展示，例如各地区的具体确诊数，新增数，某日期的新增病例数等数据可以通过将鼠标悬浮在图表指定区域上进行查看，在可视化工具的选择上，使用 Pyecharts 生成的图表具有很好的可交互性，美观的页面设计和丰富多样的图表类型使其成为数据分析和可视化的强大工具，本项目采用 Pyecharts 完成了疫情数据可视化的工作。

2.3. 根据疫情数据建立神经网络模型进行训练并预测疫情趋势

此次席卷全球的新冠肺炎是一种典型的传染病，目前，现有的新冠疫情预测模型较多，其中包括回归模型、传播动力学模型和时间序列模型等。回归模型能够比较准确地反映出时间序列指定时刻的数值，但对于早于该时刻的情况则其预测能力有限；传播动力学模型在疫情爆发早期的数据支持下可较为准确地预测疫情走势，但因为早期数据的缺失性和非动态性，在科学预测方面仍然存在问题[9]；而对于时间序列模型，其数据收集和使用相对来说就比较简单，能够很好地预测短时间内的传染病波动，但对于突然暴发的新冠疫情的预测就捉襟见肘了，因此新兴的神经网络组合模型得到了应用。并且由时间序列的原理可推论得到，随着时间的变化，新冠肺炎疫情的发展也在变化[4]，而累计确诊病例数和日累计死亡病例数等数据对疫情统计和预测都有着举足轻重的重要参考价值。于是，可以建立基于时间序列的反向传播神经网络模型，借助 MATLAB 软件进行训练和测试，对中国新冠肺炎日新增确诊病例数进行预测分析，以期进一步掌握新冠肺炎疫情发展变化的规律，为政府防控新冠疫情提供参考[10]。

3. 新冠肺炎疫情预测及可视化系统研究的意义

新冠肺炎(COVID-19)发展至 2022 年，因为其恐怖的流行性和传播性，被世界卫生组织定义为全球危险最高级的流行病。

在万物互联的今天，新冠肺炎疫情预测及可视化系统采样于国家卫健委、医疗系统的实时数据信息，利用数据收集理论形成原始数据，最后依靠前后端开发技术实现疫情动态信息显示。疫情动态系统能够第一时间为民众带来最可靠的疫情资讯，让人民群众正确深入地了解新冠病毒。信息可视化平台符合民众迫切想了解疫情的心理，能一目了然地反映出疫情的现状和走势，还能在不失真的情况下满足人们[11]。

利用疫情大数据可视化平台建立疫情预测模型，掌握疫情的发展趋势，能够尽可能地帮助政府调整政策，控制疫情规模，防止其扩散传播，甚至可能在不远的将来彻底消灭新冠肺炎，让全球人民早日从瘟疫的阴影中解脱。

4. 新冠肺炎疫情预测及可视化系统设计

4.1. 通过 Python 网络爬虫收集疫情数据信息

网络爬虫技术是指通过编写一段程序或脚本，自动获取互联网网页内容和信息的技术，爬虫可以根

据网页的统一资源定位符(Uniform Resource Locator, URL)自动获取网页的内容，返回得到网页的源代码，然后依据事先编写好的解析网页源代码的格式和内容的方法，通过解析网页内容便可获得想要获取的数据信息。其基本流程为请求网页，获取相应内容，解析内容，存储解析的数据。因为频繁的网络爬虫会给目标资源的服务器带来负担，因此很多网站也会设置相应的反爬虫机制来阻止用户的请求。

本项目通过 Python 网络爬虫获取数据集，数据源网页主要有腾讯疫情和网易疫情。腾讯疫情平台的数据主要是用于获取全国的各省市具体情况(一个月内)以及全球疫苗的接种情况。但是要做出全球的疫情情况，尤其是动态变化的过程，数据之间的联系情况，用腾讯平台的数据显然不太足够，而网易平台虽然对国内的数据并不是十分齐全，但是对于全球每一个国家从 2020 年开始的疫情数据是十分充足的，那么对于全球的疫情数据，我们主要是通过网易新闻平台进行爬取，作为对腾讯疫情数据的补充。

现在的网站数据爬取，除非一些特殊情况，我们只能通过对页面元素进行分析，对 dom 元素进行各种操作获取之外，很多的平台都会暴露出数据获取的接口，一般返回的是 json 格式的数据，只需要在浏览器的开发者视角，观察网络的发送获取情况，就可以发现暴露出来的数据接口。Python 库中的 requests 库可以对这些接口进行处理，我们对 json 数据的格式进行分析，就可以获取到我们所需要的信息，然后就可以完成由网站到 json 到数组到数据库的操作。但是现在大多数网站都有反爬机制，最常用的操作就是，构建 header，伪装一个 User-Agent 对象，将浏览者伪装成一个用户浏览器，网站发现是爬虫对象就会设置请求失败，但是构建成 User-Agent 就可以成功相应。

在腾讯疫情首页，进入开发者模式观察网络控制台接受的数据内容，在页面上随意点击一个省份查看其疫情状态，这里以广东省为例，可以看到网络控制台出现了名为 list?adCode=440000&limit=30 的 json 文件(如图 1)。

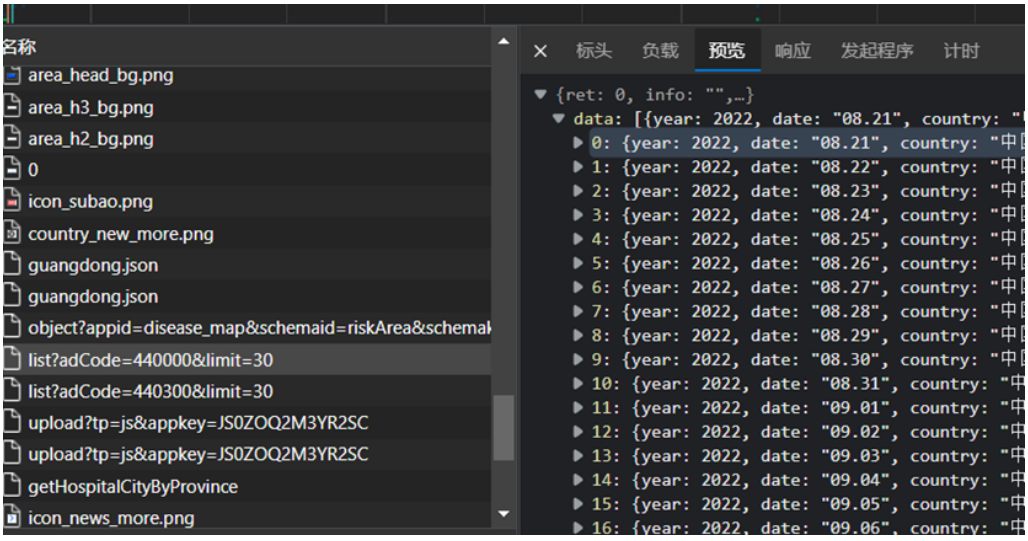


Figure 1. Tencent's json file containing epidemic data of Guangdong Province
图 1. 腾讯疫情包含广东省疫情数据的 json 文件

这里的 adCode 为中国行政区划代码，广东省的行政区划代码为 440000，limit=30 则限制了该文件记录了该省份 30 天内的疫情数据，包含了确诊数，新增确诊，死亡数，治愈数等数据。那么若要爬取全国的疫情数据，则可以此文件的网页链接为样本，依次输入各省份的行政区划代码替换 440000 组成新的网页链接，通过 request 获取到 json 文件的文本内容，然后借助 json 的语法或正则表达式获取想要得到的数据内容即可。下图为通过 RE 解析对腾讯疫情全国各省份数据爬虫的部分代码实现(图 2)。

```
def get_data():  # 腾讯疫情
    baseUrl = 'https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?adCode='
    headers = {
        "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.
    for key, value in adCode.adCode.items():
        url = baseUrl + key + "&limit=30"
        resq = requests.get(url, headers=headers)
        resq.encoding = 'utf-8'
        page_content = resq.text
        obj = re.compile(
            r'"date":"(?P<date>.*?)".*?"province":"(?P<province>[\u4e00-\u9fa5]+?)","confirm":(?P<confirm>.*?),"dea

        result = obj.finditer(page_content)
        for i in result:
            print(i.group("date") + ' ' + i.group("province") + ' ' + i.group(
                "confirm") + ' ' + i.group("dead") + ' ' + i.group("heal") + ' ' + i.group("confirm_add"))
            data = [i.group("date"), i.group("province"), i.group("confirm"),
                    i.group("dead"), i.group("heal"), i.group("confirm_add")]
            DBUtil.insert(data)
        print(value + "完成! ------------------")

    print("爬虫完成! ")
```

Figure 2. One month epidemic data of all provinces in China obtained by Tencent epidemic website crawler
图 2. 腾讯疫情网页爬虫获取全国各省一个月疫情数据

爬虫程序需要实时去更新，因为网站的数据结构在不断的变化，本项目的需求是要做到实时更新的，所以项目的爬虫软件也要更新目标网站不断作出调整，做到实时更新。

4.2. 将数据清洗后存入数据库中实现持久化

数据持久化是指将内存中的数据保存到可永久保存的设备中，最常用的做法就是将数据写入到数据库中，数据库即存储数据的仓库，是电子化的文件柜。现在的数据库系统主要分为关系型数据库和非关系型数据库，关系型数据库的存储格式可以直观地反映实体之间的关系，而且关系型数据库在数据的查询方面功能十分强大，支持结构化查询语言，即 SQL 语句，查询速度较快，常见的关系型数据库有 Mysql 和 SqlServer 等，本项目即采用 Mysql 数据库来存储疫情数据。

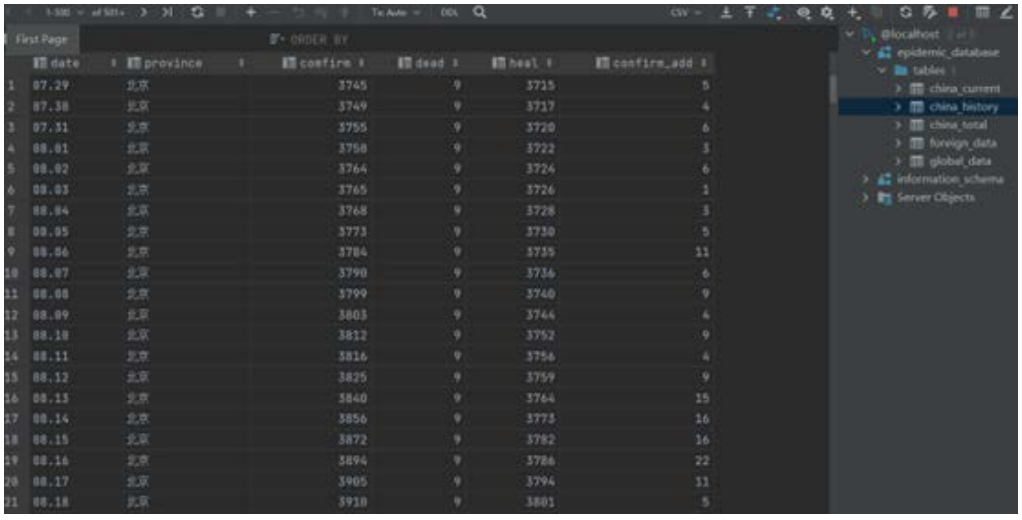


Figure 3. China_history table structure
图 3. China_history 表结构

在爬虫获得所需要的疫情数据后，我们开始建立关系型数据库，并依据这些数据内容的格式在数据库中创建特定存储结构的数据表。为方便管理，对不同爬虫模块获取的数据实行分表存储，比如，对腾讯疫情爬虫获得的全国各省一个月内的数据建立表 china_history 存储，对网易疫情爬虫所得的全球疫情数据(包含中国)建立表 foreign_data 存储，对腾讯疫情爬虫所得的全球疫情数据(不包含中国)建立表 global_data 存储。数据库的部分设计如图 3 所示。

在进行数据清洗工作时，由于爬虫网络具有较高的稳定性，所得到的数据内容已经经过计算机网络差错检测，并且疫情网页的数据一般情况下不会发送变化，因此可认为没有数据差错，脏数据的出现。但是在实际执行的过程中发现，网易疫情对于全球各国没有疫情数据的地区返回的数据类型是 None，这种类型在执行数据库插入操作的过程时会发生报错，因此需要进行数据筛选，这里定义一个函数，若数据值是 None 则返回-1，代表数据内容缺失，若不是则返回原数，如图 4 所示，实现对数据的清洗。

```
def check(data):  # 检查数据类型，处理缺失值
    if data is None:
        return -1
    else:
        return data
```

Figure 4. Define the check function to check missing values
图 4. 定义 check 函数检查缺失值

4.3. 使用爬取数据创建可视化图表

数据可视化是指借助于图形化手段，清晰有效地传达数据内容所蕴含信息。通过图表使冗长的数据表达更加形象化，可以把问题的重点有效传递给观者。常见的数据可视化图表类型有柱形图，折线图，饼状图，点图和面积图等。选择不同类型的图表可以表现出数据在特定方面的数据特征，因此要根据想要传达的数据信息使用特定的图表类型，才能发挥出数据可视化的最好效果。

为了准确地将数据进行可视化处理，我们在 python 平台调用了强大的第三方库 pyecharts。在 python 语言的支持下，百度开源的数据可视化技术 Echarts 很轻松的移植入 python 开发环境中，凭借着良好的交互性和精巧的图表设计，得到了很多开发设计人员的认可，当数据分析遇到了数据可视化时，pyecharts 就能完美完成本次项目的可视化目标。

通过数据整理后，本次项目完成了中国疫情现有确诊柱形图、中国历史新增确诊折线图、中国实时疫情确诊人数图，中国疫情预测曲线(采用神经网络模型)，中国累积确诊玫瑰饼图、新冠疫情发展趋势图和世界疫情累积确诊图。

在 pyecharts 中，所有方法均支持链式调用。链式调用是设计程序的模式，其优势是代码量大大减少，逻辑集中清晰明了，且易于查看和修改(可参考图 5)。并且在 pyecharts 中，一切皆 Options，即使用 Options 配置项来设计图表样式。

根据图表种类的不同引入不同的图表样式库。以图 7 为例，在链式调用中可定义可视化图表的宽度和高度，theme 定义了图表的主题项，add_xaxis 和 add_yaxis 分别定义了图表横纵坐标的意义，获取数据则调用 DBUtil 类中的方法，通过调用 sql 语句查询表的方法获取表内的数据，所得省份作为 x 轴坐标，

现有确诊作为 y 轴坐标，再通过 set_global_opts 定义全局变量，设置图表的题目，字体大小和区域缩放等配置项。

```
def china_Bar() -> Bar:
    bar = (

        Bar(init_opts=opts.InitOpts(width="550px", height="350px", theme=ThemeType.DARK))
            .add_xaxis(DBUtilofCC.queryProvince())
            .add_yaxis("现有确诊数", DBUtilofCC.queryConfirm())
            .set_global_opts(
            title_opts=opts.TitleOpts(title="中国疫情柱形图",
                                      title_textstyle_opts=opts.TextStyleOpts(font_size=14)),
            datazoom_opts=[opts.DataZoomOpts()],
        )
    )
```

Figure 5. Key code segments of the current confirmed column chart of China's epidemic situation
图 5. 中国疫情现有确诊柱形图关键代码段

在 pyecharts 中没有具体的标题栏设计，所以本项目在 ViewProducer 类中定义了 header 方法，实例化了 pyecharts 的 table 对象并将 rows 设为空，如此就能够将标题栏以单行表的形式返回。

由于上述图表方法运行后均生成在不同的 html 中，所以为了实现将多表组合，本项目还运用了 Page 组件，并且选用 DraggablePageLayout 方法，即拖拽的方式。将 page 的 layout 属性设置为拖拽方法并实例化后，可获得多张可拖拽图表的 html 文件(图 6)，并会自动生成能够保存图表样式的按钮 Save Config，当点击按钮后页面会自动保存图片样式并生成 json 文件(图 7)，只需将 json 文件重新加载即可获得拖拽后的 html 页面，并且该页面格式会固定无法更改(图 8)。

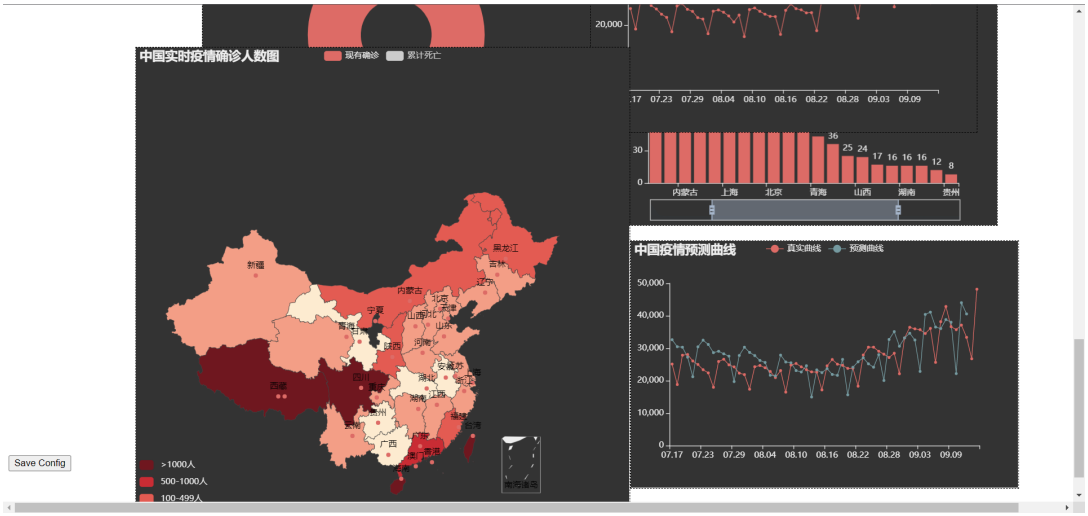


Figure 6. Chart style before dragging
图 6. 拖拽前图表样式

Figure 7. Json style file
图 7. Json 样式文件

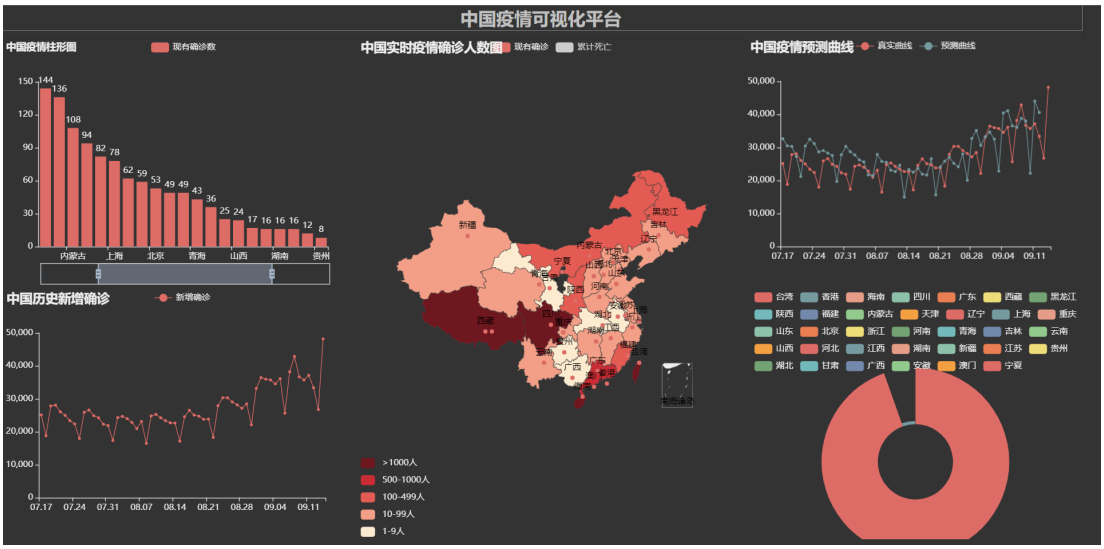


Figure 8. Chart style after dragging
图 8. 拖拽后图表样式

4.4. 训练神经网络模型实现对疫情曲线的预测

某地区每日的疫情数据所组成的数据集是一种典型的时间序列，对时间序列进行分析和预测常用的神经网络模型有循环神经网络(Recurrent Neural Network, RNN)模型和长短期记忆网络(LSTM, Long Short-Term Memory)模型等，LSTM 模型就是 RNN 模型的一种衍生品，具有“门”结构，通过门的逻辑控制数据的更改或舍弃，克服了以往 RNN 模型的权重影响太大，造成过拟合和梯度消失的问题，能够提高网络的预测精度，实现网络的收敛。本项目采用 LSTM 模型对疫情数据集进行训练和预测。

4.4.1. LSTM 神经网络模型结构与原理

LSTM 拥有三个门，依次分别是信息遗忘门、输入记忆门、输出记忆门，以此来决定每隔一时刻的信息记忆和遗忘。输入门是确定会有一个什么时候新产生的讯息会进入到细胞当中，遗忘门是控制每一时刻的讯息输入是否都会自动被遗忘，输出门是确定在每一时刻是否会有新讯息输出。细胞状态是 LSTM 的核心内容，用贯穿细胞的水平线表示。像传送带一样，细胞状态贯穿整个细胞并且没有多余的分支，以此保证信息不变地通过整个循环神经网络模型。细胞状态如图 9 所示：

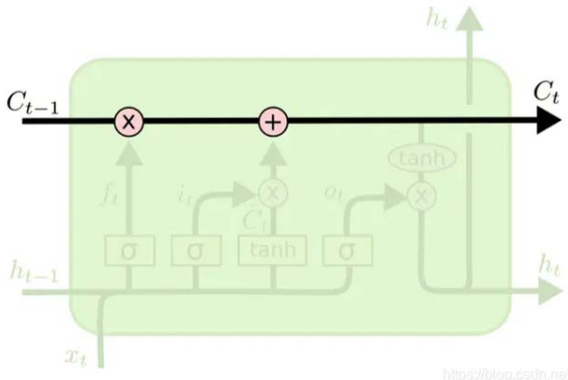


Figure 9. LSTM cell status chart
图 9. LSTM 细胞状态图

LSTM 网络中最重要的组成部分是门单位。门能够选择性地过滤信息。门的结构如图 10 所示：

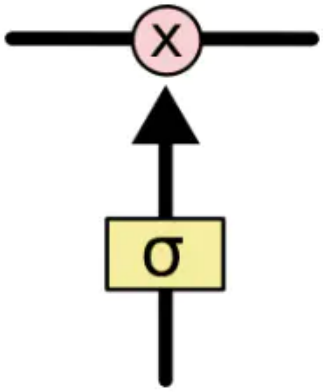
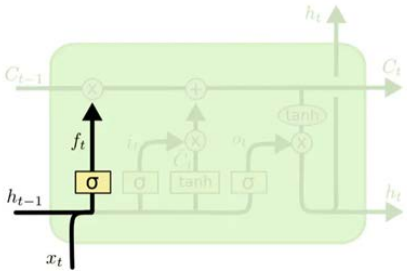


Figure 10. Structural diagram of LSTM door
图 10. LSTM 门结构图

sigmoid 层输出 0-1 的值，能够控制进入 sigmoid 层的信息量。数值为 0 不能通过，数字为 1 则能通过。每个 LSTM 单元中含有 3 个控制细胞状态的门单元，即遗忘门，input 门和 output 门。

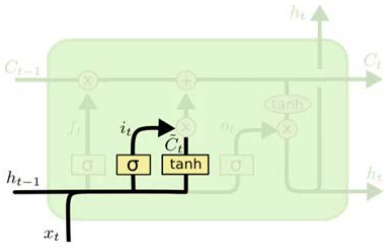
遗忘门：即一个 sigmoid 单元，它负责处理信息并决定细胞状态需要遗忘即丢弃哪些信息。具体工作原理是：其通过查看 h_{t-1}和 x_t 的信息来输出一个值位于 0-1 之间的向量，该向量的 0-1 值表示细胞状态 C_{t-1}中的哪些信息丢弃或保留多少。0 表示都不保留，1 表示都保留。遗忘门的结构如图 11 所示：



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 11. Structure of LSTM forgotten door
图 11. LSTM 遗忘门结构图

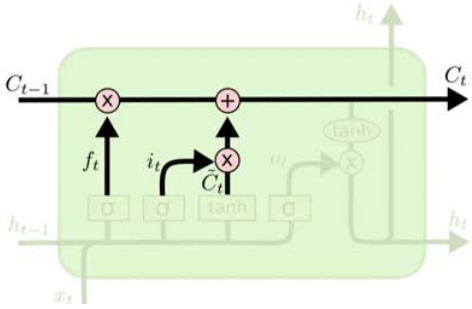
输入门：然后是决定给细胞状态更新哪些新的信息。这一步又分为两个步骤，首先，利用 h_{t-1}和 x_t 通过一个称为输入门的单元的操作来决定更新哪些信息。然后利用 h_{t-1}和 x_t 通过一个 tanh 层得到新的候选细胞信息{C_t}~，这些信息可能会被更新到细胞信息中。这两个步骤如图 12 所示：



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] \; + \; b_C\right)$$

Figure 12. LSTM cell update action chart (1)
图 12. LSTM 细胞更新动作图(1)

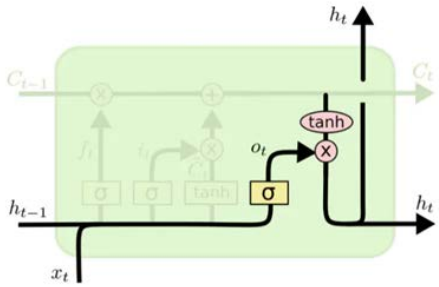
然后将更新细胞信息 C_{t-1}。更新的规则就是通过遗忘门选择遗忘旧细胞信息中需要遗忘的部分信息，通过输入门选择添加候选细胞信息{C_t}~中需要添加的部分信息得到新的细胞信息 C_{t}。更新操作如图 13 所示：



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 13. LSTM cell update action chart (2)
图 13. LSTM 细胞更新动作图(2)

输出门：更新完了细胞状态信息后，还需要根据输入的 h_{t-1}和 x_t 来判断输出细胞状态的哪些状态特征，可以通过将输入的 h_{t-1}和 x_t 经过一个称之为输出门的 sigmoid 层得到一个判断条件，然后将细胞状态经过 tanh 层得到值位于一个−1~1 之间的向量，该向量与输出门所得到的判断条件相乘就得到了最终该 LSTM 单元的输出。该步骤如图 14 所示：



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 14. Structure of LSTM output gate
图 14. LSTM 输出门结构图

4.4.2. 处理疫情数据集

通过之前的网络爬虫和数据持久化过程，项目的数据库中已经存放了最新的疫情数据，其中历史数据有中国各省和中国全境的一个月内的数据，这里以中国全境的历史数据为例，进行数据处理方便 LSTM 模型进行训练。过程包括数据格式转换和数据归一化等操作。代码部分如图 15 所示。

首先通过 DBUtil 工具类定义的查询操作得到中国全境一个月内的每日新增确诊数数据，返回的格式为一个整型列表，然后以其为参数借助 pandas 库的 DataFrame 构造方法和 values 变量返回一个 Numpy 数组 dataset 实现格式的转换，然后利用 Numpy 的 max 和 min 函数获取该数据集的最大值和最小值，两者相减得到极差 scalar，最后通过匿名函数将数据集中的每个数据除以 scalar 实现数据归一化。

另外，定义一个处理数据集的函数，我们对模型进行训练时要有两个数据集，一个是输入数据集，另一个是目标数据集。在这里我们将该疫情历史数据进行划分，以前三个数据作为后一个数据的输入，以此类推，得到两个长为 len-3 的数据集(len 为历史数据长度)，其中输入数据集的每个元素由三个数据组成。

```
# LSTM (Long Short-Term Memory) 是长短期记忆网络
data = DBUtilofCT.queryConfirmAdd()

plt.plot(data)
plt.show()
# 数据预处理
# data = data.dropna()  # 去掉na数据
# dataset = data.values  # 字典(Dictionary) values(): 返回字典中的所有值。
dataset = pd.DataFrame(data, columns=['confirmAdd']).values
dataset = dataset.astype('float32')  # astype(type):实现变量类型转换
max_value = np.max(dataset)
min_value = np.min(dataset)
scalar = max_value - min_value
dataset = list(map(lambda x: x / scalar, dataset))  # 将数据标准化到0~1之间


def create_dataset(dataset, look_back=3):  # look_back 以前的时间步数用作输入变量来预测下一个时间段
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back):
        a = dataset[i:(i + look_back)]  # i和i+1赋值
        dataX.append(a)
        dataY.append(dataset[i + look_back])  # i+2赋值
    return np.array(dataX), np.array(dataY)  # np.array构建数组
```

Figure 15. Epidemic data set processing
图 15. 疫情数据集处理过程

4.4.3. 利用疫情数据集进行 LSTM 模型的训练

训练过程如图 16 所示：

```
for e in range(10000):
    var_x = Variable(train_x)  # 转为Variable（变量）
    var_y = Variable(train_y)

    out = net(var_x)
    loss = criterion(out, var_y)

    optimizer.zero_grad()  # 把梯度置零，也就是把loss关于weight的导数变成0.
    loss.backward()  # 计算得到loss后就要回传损失，这是在训练的时候才会有的操作，测试时候只有forward过程
    optimizer.step()  # 回传损失过程中会计算梯度，然后optimizer.step()根据这些梯度更新参数
    if (e + 1) % 100 == 0:
        print('Epoch: {}, Loss:{:.7f}'.format(e + 1, loss.item()))

torch.save(net.state_dict(), 'net_params.pkl')  # 保存训练文件net_params.pkl
# state_dict 是一个简单的python的字典对象,将每一层与它的对应参数建立映射关系
```

```
Python Console × train (1) ×
Epoch: 9300, Loss:0.0000160
Epoch: 9400, Loss:0.0001357
Epoch: 9500, Loss:0.0000105
Epoch: 9600, Loss:0.0000041
Epoch: 9700, Loss:0.0000184
Epoch: 9800, Loss:0.0000178
Epoch: 9900, Loss:0.0000305
Epoch: 10000, Loss:0.0000167
```

Figure 16. LSTM model training
图 16. LSTM 模型训练

我们进行了 10,000 轮次的循环训练，每 100 轮输出 loss 的值反映训练进度，可以看到在进行了 10,000 轮训练后，损失函数的值已经非常小。最后我们把训练好的模型参数保存下来，用于后面的测试。

4.4.4. 利用训练后的 LSTM 模型进行测试

我们利用训练好模型进行测试时，前 70%的数据集为训练集，后 30%的数据集为测试集，根据训练好的模型参数，我们定义预测函数实现对中国历史新增病例数的预测，并将输入数据和输出数据借助可视化工具展现在图表上，输出的疫情预测曲线图如图 17 所示。模型的 output 与训练样本的 output 高度匹配，但是测试集依然有不小的误差，这是由于数据集的数据太少，当我们增加训练数据集的数据量，模型训练的效果会更加符合真实情况。

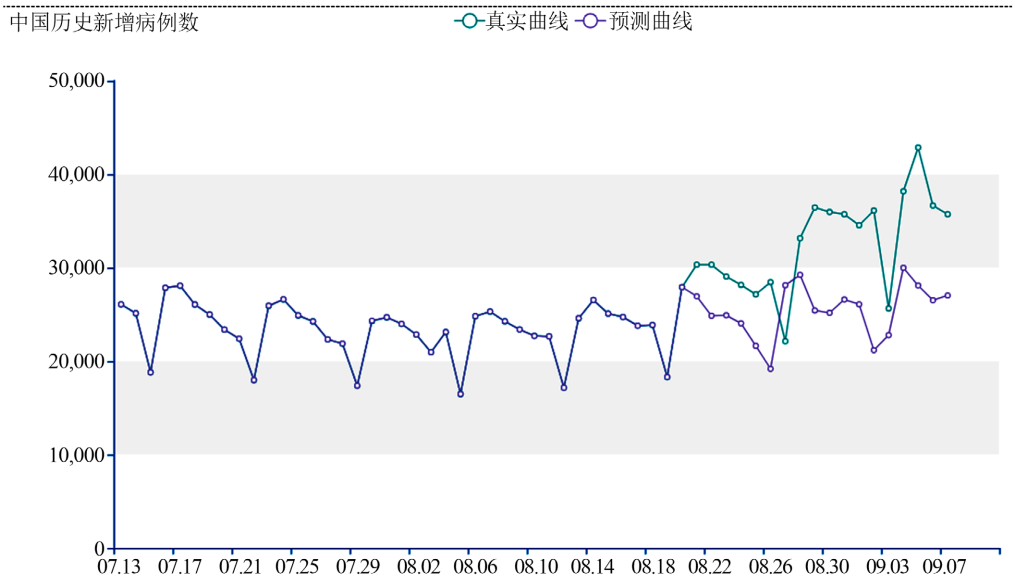


Figure 17. Forecast of historical increase in China
图 17. 中国历史新增预测图

5. 结论

本文分析了新冠肺炎疫情预测及可视化系统的国内外研究背景和研究现状，在可行的基础上确定了本项目的研究内容，包括网络爬虫获取疫情数据，数据持久化，数据可视化，建立神经网络模型进行数据预测等，并结合当前疫情状况分析出该项目的研究意义：旨在借助数据可视化技术将蕴藏在海量数据中的疫情信息直观地表现出来，为疫情防控工作提高便利，以期早日消灭疫情。最后在系统设计方面给出具体方案，包括网络爬虫模块的网络控制台搜索数据源网页，正则表达式解析数据内容等，数据持久化模块的数据库设计，数据清洗等，可视化模块的基本图表生成，配置文件存储等，疫情预测模块的 LSTM 原理分析，神经网络训练，测试，生成预测曲线等。通过上述方案的实施，我们完成了该系统的设计。

本项研究还有以下待改进的地方：

1) 收集所得的数据内容不够完整，部分网页爬虫所得的全球疫情数据有一些地区是未公布的，这些缺失值未得到有效处理，可以扩大数据源网页的搜索范围进行补充。

2) 收集所得的数据内容不够全面，除了新增确诊数，死亡数，治愈数，累计确诊数等，一些其他数据比如高风险地区数，死亡率，治愈率，接种疫苗数等也可以加入到收集的数据集中。

3) 数据可视化的图表类型不够丰富，数据可视化工作实现了柱形图，折线图，玫瑰图，疫情地图等基本图表的展示，其他类型的图表有待补充。

4) 预测模型效果不够理想，疫情预测模块的测试工作里我们发现疫情预测曲线虽与真实曲线大致相似，但依然有些误差，可以通过扩大训练数据集来提高预测效果。

致　谢

感谢北京信息科技大学大学生创新创业训练计划项目的资金支持，感谢张翠平老师的指导。

基金项目

基金资助由北京信息科技大学大学生创新创业训练计划项目－计算机学院(5112210832)支持。

参考文献

[1] 艾廷华. 大数据驱动下的地图学发展[J]. 测绘地理信息, 2016, 41(2): 1-7.

[2] 曾悠. 大数据时代背景下的数据可视化概念研究[D]: [硕士学位论文]. 杭州: 浙江大学, 2014.

[3] 邢策梅, 陶金梅, 范娟娟, 瞿明霞. 基于微服务架构的江苏省新冠疫情大数据平台的设计与实现[J]. 现代测绘, 2020, 43(3): 34-38.

[4] 王旭艳, 喻勇, 胡樱, 宇传华. 基于指数平滑模型的湖北省新冠肺炎疫情预测分析[J]. 公共卫生与预防医学, 2020, 31(1): 1-4.

[5] 盛华雄, 吴琳, 肖长亮. 新冠肺炎疫情传播建模分析与预测[J]. 系统仿真学报, 2020, 32(5): 759-766. https://doi.org/10.16182/j.issn1004731x.joss.20-0156

[6] 曹盛力, 冯沛华, 时朋朋. 修正 SEIR 传染病动力学模型应用于湖北省 2019 冠状病毒病(COVID-19)疫情预测和评估[J]. 浙江大学学报(医学版), 2020, 49(2): 178-184.

[7] 任晓龙, 李忠, 申天恩, 毛亦鹏, 宋俊杰. Logistic 回归模型对美国新冠疫情预测研究[J]. 福建电脑, 2021, 37(4): 47-49. https://doi.org/10.16707/j.cnki.fjpc.2021.04.011

[8] 李昊, 段德光, 陶学强, 陈恩, 高树田. 传染病动力学模型及其在新型冠状病毒肺炎疫情仿真预测中的应用综述[J]. 医疗卫生装备, 2020, 41(3): 7-12.

[9] 江海峰, 胡根华, 梅昱楠. 我国 COVID-19 日新增确诊病例存在“泡沫”行为吗? [J]. 安徽工业大学学报(自然科学版), 2021, 38(1): 104-110.

[10] 吴琦琦, 黄志甲, 周恒, 卞梦园, 寇遵丽, 张金星. 基于时间序列神经网络的新冠肺炎疫情预测[J]. 安徽工业大学学报(自然科学版), 2021, 38(2): 188-194.

[11] 吴辛迪, 吴冬原, 郑凯明. 信息可视化在界面设计中的应用研究——以“新冠肺炎疫情实时动态”系统界面为例[J]. 设计, 2020, 33(8): 93-95.