

省级农作物病虫害数据上报系统的设计与实现

王小宁¹, 王儒敬^{1,2}, 桂元苗^{1,2}, 王雪¹, 宿宁¹, 魏圆圆^{1*}

(1. 中国科学院合肥智能机械研究所, 合肥 230031; 2. 中国科学技术大学信息科学技术学院, 合肥 230026)

摘要: 针对省级农作物病虫害测报业务的需求, 围绕“省、市、县”三级植保体系, 设计开发了省级农作物病虫害数据上报系统。系统开发采用浏览器/服务器(B/S)架构, Java作为开发语言, 数据库使用MySQL数据库。系统后端使用Spring Boot框架搭建, 前端基于Vue数据绑定和D3的可视化技术, 实现了省级农作物病虫害调查日报、周报、年报等报表数据的上报、可视化分析, 支持PC端和移动端通过浏览器对系统的访问。系统通过初步的测试和试用, 满足目标用户需求。

关键词: 农作物; 病虫害测报; 植物保护; Spring Boot; Vue; D3

中图分类号: S126; S431

文献标识码: A

文章编号: 1672-352X(2019)05-0876-07

Design and implementation of provincial crop pest and disease data reporting system

WANG Xiaoning¹, WANG Rujing^{1,2}, GUI Yuanmiao^{1,2}, WANG Xue¹, SU Ning¹, WEI Yuanyuan¹

(1. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031;

2. School of Information Science and Technology, University of Science and Technology of China, Hefei 230026)

Abstract: Aiming at the demand of provincial crop pests and diseases reporting business, the paper designs and develops a provincial crop pest and disease data reporting system around the “provincial, municipal and county” three-level plant protection system. The system development adopts the browser/server (B/S) architecture, Java as the development language, MySQL as the database. The system backend is built using the Spring Boot framework, and the front end is based on Vue framework and D3 visualization technology. It achieves the reporting and visual analysis of report data such as daily crop, pest and disease surveys, including the daily, weekly and annual reports. This system supports PC and mobile accesses through the browser. The system meets the needs of target users through preliminary testing and trials.

Key words: crop; pest measurement and forecast; plant protection; Spring Boot; Vue; D3

现代植保体系是为了适应社会经济和生态保护的总体要求, 将服务现代农业作为任务驱动力, 基于现代科技、现代装备、现代人才以及政策保障的支撑手段, 所建立的可持续治理农作物病虫害的新型农业防灾减灾体系^[1]。2000年以来, 在党和政府的高度重视下, 全国各级植保部门抓住信息化快速发展的历史机遇, 在新型测报工具研发应用、信息系统建设、预报发布方式创新等方面取得了明显进展, 其中“加快信息平台建设, 推进测报手段信息化”仍将是今后较长一段时间里的重点任务之一^[2]。

在病虫害测报信息系统建设方面, 全国农业技术

推广服务中心开发建成的“全国农作物重大病虫害数字化监测预警系统”已投入运行近10年, 初步实现了省级重大病虫害测报数据向农业农村部的网络化报送^[3-6]。同时, 各省级植保机构开发建设了各具特色的病虫害测报信息系统, 实现所辖县市病虫害数据的填报和分析处理^[7-14]。

当前, 互联网、大数据和云计算等信息技术飞速发展, 迫使加快系统开发技术的革新; 种植业产业结构调整、病虫害发生态势的日益变化, 使得在系统扩展性和系统维护等方面都存在瓶颈。在日常与安徽省农委植保部门工作接触中, 笔者也深刻感觉

收稿日期: 2019-02-22

基金项目: 国家自然科学基金项目(31401293), (31671586)和(61773360)共同资助。

作者简介: 王小宁, 硕士研究生。E-mail: 755292832@qq.com

* 通信作者: 魏圆圆, 博士, 副研究员。E-mail: jsjwyy@126.com

到各级植保机构对现有平台升级换代的强烈诉求。

同时在进行数据可视化分析时,也发现仅从病虫害的角度进行可视化,难以表达区域层面的对比效果。原因在于通过传感器只能够获取监测点上的数据,在进行较大尺度的区域监测时,基于“点”的物联网监测技术就会面临数据空间不连续、区域性分析困难等诸多难题^[15-16]。地图可视化可弥补分析人员面对大量冗余表格数据、工作单调的缺点,能利用地形地势资源、天气气象动态,在不同的精度(省、市、县)下进行空间数据的汇总、显示、编辑和制图。所以对信息可视化、图形多样化的需求也应运而生,WebGIS^[17-19]、百度谷歌地图 API 服务、Echarts、D3^[20-24]等功能强大、样式丰富的技术得到越来越多的使用。

本文以构建普适性强、扩展性好、易学易用的通用平台为研究目标,针对省级农作物病虫害数据上报系统的业务需求,结合安徽省病虫害测报体系的数据上报内容,设计并开发了一套省级农作物病虫害数据上报系统。系统开发采用浏览器/服务器(B/S)架构,Java 作为开发语言,数据库使用 MySQL 数据库,系统后端使用 Spring Boot 框架搭建,前端基于 Vue 数据绑定和 D3 的可视化技术,构建面向“省、市、县”三级的病虫害数据上报、审批的病虫害监管体系,实现了省级农作物病虫害调查日报、周报、年报等报表数据的上报、可视化分析,支持 PC 端和移动端通过浏览器对系统的访问。

1 系统设计

1.1 需求分析

本系统针对现有病虫害上报系统开发技术老旧、审批机制不健全,系统可维护性、扩展性越来越不能满足用户需求等问题,以构建普适性强、扩展性好、易学易用的省级通用平台为目标,围绕“省、市、县”三级植保业务体系中病虫害数据在线填报、在线审批等功能需求展开需求分析和系统设计,采用当下流行的微服务框架搭建,在系统可维护性和扩展性方面都有着一定的提升,并且实现了完善的在线审批功能。同时针对从大量表格数据中,无法方便获取直观的对比分析结果的缺陷,系统提供以下可视化统计分析功能:数据列表对应的统折线图、不同地区不同年份不同作物病虫害数据的堆栈图展示,和以颜色深浅表示病虫害发生程度的区域可视化分级显示。

1.2 业务逻辑关系

综合收集的数据资料,整理生产管理过程中的

具体业务需求,在业务逻辑上,围绕植保机构(县级、市级和省级)、作物病虫害类型、报表类型 3 个维度展开。一方面各级植保站填报、维护本站数据,同时上级植保站具有审核下级植保站提交数据的权限。以安徽省为例,肥东县植保站将病虫害发生数据上报至其上级合肥市植保站;合肥市植保站对所辖县植保站上报的数据进行审核,同时也可以上传市级的数据到安徽省植物保护总站。下级植保站填报数据时,未选择上报则该条数据保存为“未提交”状态,若选择上报则将数据发送到上级植保站,在未得到审核结果之前,该条数据处于“灰度”不可编辑状态。上级植保站审核时有两种处理结果:通过和未通过,未通过时可以填写未通过原因返回给下级植保站,通过后报表保持为不可修改状态。病虫害报表类型按照上报时间频率,分为日报、周报、年报。报表的审核业务流程如图 1 所示。

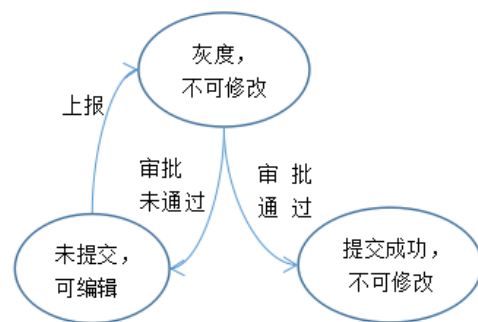


图 1 报表审核业务流程

Figure 1 Report escalation business process

作物病虫害数据依据安徽省农作物的主要品种及其所发生的主要病虫害,初步涵盖小麦、水稻、大豆、油菜、玉米等大宗农作物。

1.3 数据库设计

省级农作物病虫害数据上报系统的报表平台使用 MySQL 关系型数据库,数据库引擎选择了默认的 InnoDB 引擎,支持 ACID 事务,保障出现高并发问题或系统故障时的数据完整性和安全性。依据前端功能模块将数据表分为 3 个类别:①基础数据,以“bs_”为前缀,包括作物、病虫害、生态区、植保站、设备、用户及用户权限等表格;②会商数据,以“cr_”为前缀,包括重大病虫害越冬基数调查、小麦中后期主要病虫害发生统计、夏秋农作物主要病虫害发生统计表;③常规数据,按次报、周报、年报分别以“tr_”、“wr_”、“yr_”为前缀,每种作物的每种病虫害都有一张表格与之一一对应,不同的前缀可以让一目了然的明白它的类别。数据表间的关系如图 2 所示。

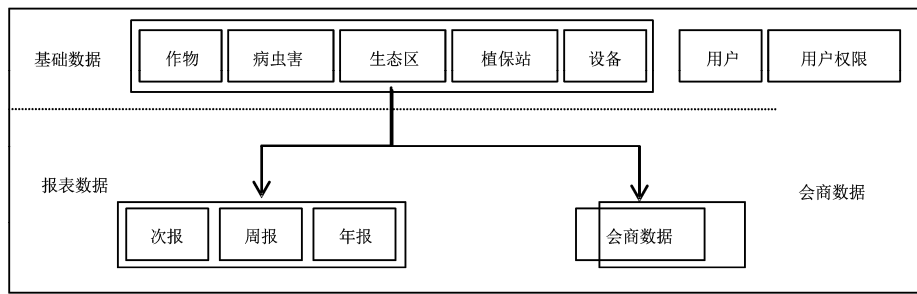


图 2 数据库表间关系

Figure 2 Database table relationship

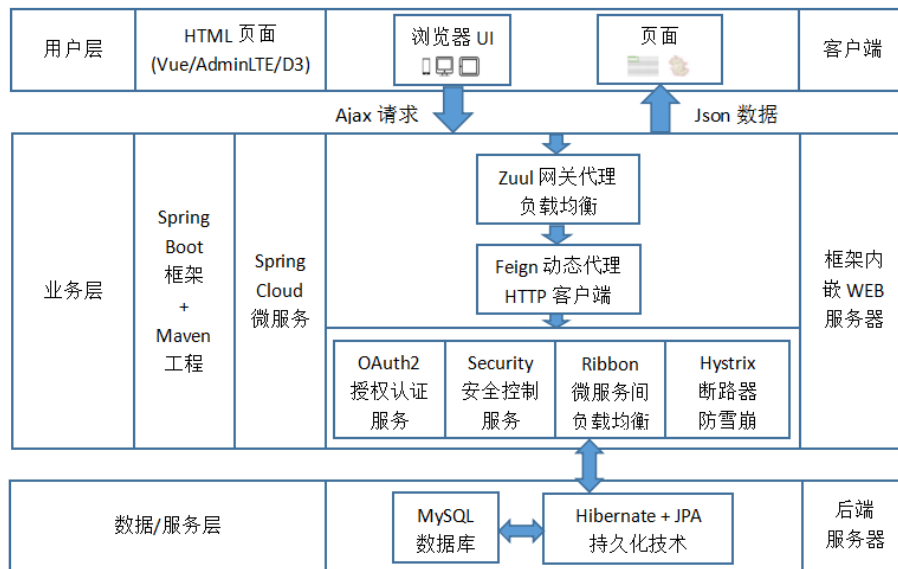


图 3 系统开发架构图

Figure 3 System development architecture

1.4 系统架构设计

系统服务端以 Spring Boot 框架为核心^[25-26], 结合 Spring Cloud 微服务架构^[27-28], 对外提供符合 RESTful 规范的接口^[29-31], 与前端的通信通过返回标准的 json 数据完成; 前端使用 Vue 2.0 框架, 通过 ajax 请求访问后端接口。系统开发架构设计图如图 3 所示。

2 系统实现

Spring Boot 框架的搭建依托于创建 Maven 工程, 按照功能的不同分别创建了 Eureka 子工程、Router 子工程、Service 子工程、Api 子工程和 Web 子工程等五个子工程, 分开部署有利于微服务的“分而治之”, 降低耦合性的同时也加快了系统从简单到复杂的开发过程, 同时有利于后期对各项微服务功能的测试以及快速的错误定位。

2.1 前端功能模块划分

Web 子工程为前端项目, 是系统的视图层, 包括前端页面及对应后台的处理流程解析, 前端业务

按功能模块分为数据填报模块和可视化模块两类。数据填报模块包含基础数据页面, 常规上报页面和会商数据填报页面, 覆盖所有报表业务的增删改查及上报审批功能; 可视化模块实现了将报表数据转化为折线图的显示, 和将病虫害发生程度的区域分布情况进行地图可视化显示并实现系列的前端地图操作。

2.2 前端功能模块实现

前端的页面功能主要基于 html、jquery、vue、adminLTE、D3 等技术实现, 前端框架和界面的构建主要依托于 vue 和 adminLTE 两大技术。

vue 是一套构建用户界面的渐进式框架, 只关注视图层, 使用 model-view-viewmodel (MVVM) 的开发模式将页面视图数据和 DOM 元素进行双向绑定, 可以比作计算机桌面图标的快捷方式, 更新一处数据 (或属性) 会自动完成另一处属性 (或数据) 的同步更新。它和 jquery 的明显不同之处就在, 修改页面数据时不需要再先获取 DOM 元素即可完成。

AdminLTE 是一款基于 bootstrap 的后台管理系

统的通用模板 UI, 它的样式美观且较为符合大多数后台管理系统的需求, 典型的上|左右|下的布局形式, 并且提供了一整套开发的时候可能用到的 UI 样式, 比如表格, 表单, 图表, 日历, 登录注册错误页面等, 非常适合对样式编排不太擅长的后端开发者, 而且实现了基本的动态交互能力, 比如时间框的交互, 密码空缺、邮箱日期等格式的正则校验, 几乎能够满足所有情况下的页面基本要求, 通过拷贝模板源代码即可应用于本地显示, 加上自定义的功能及样式修改即可快速使用。有了便可以不用自己去写很多复杂的 html 和 css, 可以把更多时间和精力留在后台的开发上。

2.3 后端功能模块划分

后端按工程功能分为 Eureka 工程、Router 工程、Service 工程和 Api 工程四类。所需的微服务依据服务的功能不同分别部署到 Eureka、Router、Service 工程, Api 工程接收前端请求, 完成数据持久化操作。

Eureka 的功能是服务注册和发现, 目的在于处理收到的访问请求并转发到正确的服务器端口 (或集群)。Eureka 是客户端/服务端 (C/S) 的结构, 每个部署了微服务的客户端都将自己的 ip 及端口注册到服务端维护的一张路由信息表, 即服务注册中心中。每当有新连接产生时, 客户端向注册中心查询当前操作应该访问的服务器端口并向得到的路径发起请求。

Router 工程负责对由 Eureka 转发的请求进行路由处理, 使用了开源的微服务网关 zuul 技术。前端请求会先到达网关, 根据不同请求的特点, 转发给对应的服务处理。ribbon 和 hystrix 是功能组件。ribbon 的作用类似 nginx, 不同之处在于 nginx 是针对外部请求的负载均衡, zuul 网关也具有此特点, 而 ribbon 是服务器得到请求之后在内部调用不同微服务时的负载均衡。如果要请求的服务部署到了多个机器或多个端口上, ribbon 会对 ServerList 列表中的可用端口进行轮询, 这样就可以把请求均衡的分配到各个端口, 从而解决“单点问题”; hystrix 是断路器, 它对不同的服务划分了线程池, 对应处理不同的服务。当某服务发生雪崩时, hystrix 会隔离、熔断该模块来保证其他模块的可用, 以及降级处理, 即先在数据库中记录业务过程, 等服务可用时再进行处理。

Service 工程是系统的业务层, 定义了平台提供的服务, 有授权认证服务 OAuth2、安全控制服务 Security 和动态代理服务 feign。其中 feign 对由 Router 工程转发的请求作进一步处理, 负责跟指定

的服务从建立连接到解析响应的所有步骤, 根据请求地址中的注解和输入参数, 动态生成需要访问的服务路径。feign 基于 HTTP 客户端, Spring Cloud Netflix 的微服务都是以 HTTP 接口的形式暴露的, 即发布到注册中心的服务方接口都是 HTTP 的。授权认证服务流程: 在认证的类文件中通过 @EnableAuthorizationServer 注解声明一个认证服务器客户端, 首次登陆会重定向到授权允许页面, 同时携带返回的 code 值, 该页面通过授权后会使用 code 值来请求 token 服务获取 token, 此后的请求头都会携带该 token 信息, 被服务器验证正确身份; 安全控制服务提供一个过滤器拦截所有的请求, 并判断是否有对该资源的访问权限, 如果无权限则重定向到登录页面。

Api 工程是系统的实体层, 负责与数据库的连接、映射及 CRUD 操作, 配置文件中集成了 JPA、Hibernate 持久化框架。该工程定义了所有数据的实体类, 每个类文件对应数据库中一张表, 并且定义了操作数据的接口方法。Spring Data Rest 会分析实体之间的关系, 生成 Rest 风格的接口。

2.4 后端功能模块实现

Spring Boot 是当前 Spring 社区一个非常热门的框架, 它的目的是为了简化 Spring 应用的开发。Spring Boot 有一个非常出色的特点“免配置搭建”, 只需很少的配置, 因为用户习惯的绝大多数的配置方法都已经被框架集成为默认配置。通过 Spring Boot 能够迅速的搭建一套 web 项目。Spring Boot 除了高度集成封装了 Spring 一系列框架之外, 还封装了 Web 容器, 在发布项目时就省去了将工程部署到 tomcat 等容器的步骤, 它会根据配置文件中声明的 web 容器属性, 在 Application 主类的 main 方法运行时实现启动容器的要求。

Spring Cloud 是一个微服务框架, 基于 Spring Boot 来开发, 封装了多个开源组件包括 Netflix 核心技术, 为开发者提供了快速构建分布式系统的工具。当项目中想使用某一种微服务时, 只需在配置文件中添加对应的依赖, maven 工程会自动下载对应版本的 jar 包, 同时只需通过 @SpringBootApplication 注解配置服务主类启动该服务即可使用。

2.5 实现要点

2.5.1 权限管理 平台对角色提供 5 种权限: ROLE_ROOT_ADMIN、ROLE_ADMIN、ROLE_DEVELOPER、ROLE_DATA_REPORTER、ROLE_USER。系统首页的功能菜单项具有对应的权限限制, 不同权限用户可看到不同菜单项。如表 1 所示。

表 1 权限和目录可见性的关系
Table 1 Permission and directory visibility

目录 Catalogue	ROLE_ROOT_ ADMIN	ROLE_ ADMIN	ROLE_ DEVELOPER	ROLE_DATA_ REPORTER	ROLE_USER
用户、机构管理	√		√		
作物、病虫害管理	√	√	√		展示页
常规上报	√	√	√	√	
会商数据	√	√	√	√	
地图可视化	√	√	√	√	

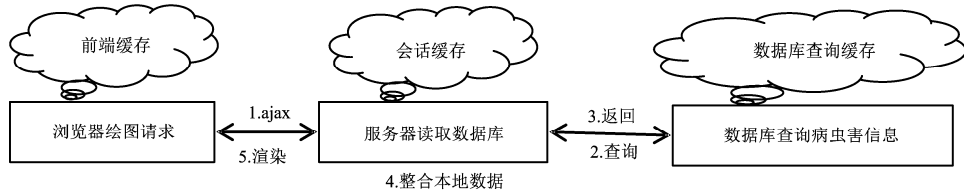


图 4 地图可视化技术实现

Figure 4 Map visualization technology implementation



图 5 【基础数据】管理菜单项

Figure 5 Basic data management menu item



图 6 【作物管理】列表界面

Figure 6 Crop management list interface

2.5.2 数据填报 平台前端样式使用 adminLTE 模板设计，具有可移植性、可重用性强的特点。数据填报页面基于 Vue 框架，页面数据和数据源模型绑定，通过请求接口获得需要的数据，非常方便实现增删改查功能。此外，对填写数据的格式、量纲、最大最小值，都会有前端提示和实时校验反馈。

2.5.3 可视化 图 4 是运用 D3 绘制图形的技术路线。当前端用户点击某作物病虫害某年份的绘图命令时，会先查询前端缓存 LocalStorage 中是否已存在所需要的缓存数据，若存在则不需再请求服务器，直接进行绘图，若前端缓存中不存在，则发起 Ajax 请求到服务器查询数据库；数据库缓存 Hibernate Cache 检查是否存在该查询语句的调用记录，存在则直接返回查询结果，若不存在，则需建立读表操作。后台得到的数据需要去冗，整合地图数据处理成前端需要的格式返回。同时最终数据也作为数据库缓存、前端缓存存储起来，从而提高缓存有效期

内的页面响应时间，提升用户的体验。

3 系统功能及应用

系统功能主要分为基础数据管理、常规数据上报和会商数据填报，以及数据图形化分析等。

3.1 基础数据管理

基础数据管理包括：作物管理，病虫害管理、测报设备管理、字典管理。超级管理员具有生态区管理、机构管理和用户管理的权限。菜单项如图 5 所示，以“作物管理”为例，图 6 是作物管理列表界面，包括数据的增删改查功能。

3.2 常规及会商数据上报

常规上报包括病虫害数据的日报（病虫害发生情况按次上报）、周报（病虫害发生时段内按周上报）、年报（病虫害发生情况年度报表填报），界面如图 7 所示。

选择某一病害或虫害，进入详细上报列表页面，如图 8 所示，上面部分是已经添加的数据索引，点开左侧按钮，可查看数据详情，界面下面部分是当

前数据列表对应的统计图, 以直观展示调查指标的变化趋势。可通过加号图标, 添加调查数据, 如图 9 所示。添加且上报的数据, 显示“未审核”状态,

此状态下不能对该数据执行编辑或删除操作; 添加但未提交的数据, 显示“未提交”状态, 此状态下可以执行编辑或删除操作。



图 7 常规上报界面

Figure 7 Regular reporting interface



图 8 水稻白叶枯病日报上报列表

Figure 8 Rice bacterial blight daily reporting list

图 9 水稻白叶枯病日报填报界面

Figure 9 Rice bacterial blight daily report interface

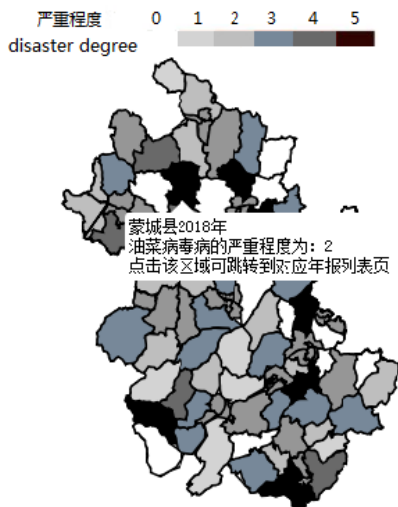


图 10 油菜病毒病发生程度分布图

Figure 10 Distribution map of rapeseed virus disease

3.3 病虫害发生程度区域分布图和堆栈图生成

将安徽省所辖各市县的不同农作物病虫害等数据绑定到地图, 在一张地图上进行可视化显示, 如图 10 所示, 不同的市县区域按发生程度的不同分为五种灰度, 由浅到深表示病虫害灾害的发生程度由轻到严重。

不同病虫害的发生程度均有其成熟或约定的判断方法, 可在地图上进行展示。地图支持缩放、拖动等操作, 鼠标悬浮在某区域上方时会及时显示该区域的作物属性, 同时点击该区域会跳转至对应的年报列表页面。

根据所选的年份和作物品种, 将地图的平面信息和第三维度的浮层数据进行汇总, 绘制出统计堆栈图, 即堆积柱形图, 方便进行各县市不同病虫害

发生程度的直观对比。

4 结论

本研究针对省级农作物病虫害数据上报系统的业务需求,构建了“省、市、县”三级植保数据上报体系,结合安徽省病虫害测报体系的数据上报内容,详细介绍了一个省级农作物病虫害数据上报系统从需求分析到开发实现的完整流程,对开发过程中使用的框架、遇到的问题、技术路线的分析,都做了较细致阐述。系统对各类病虫害发生数据进行结构化管理基础上,对病虫害发生程度分布的地理信息进行可视化交互,同时提供统计分析功能。通过在数据报表分析中引入可视化技术,对数据进行多样化的直观表达,帮助支部人员快速统计、分析和做出决策。目前平台通过了测试和初步试用,较好地满足目标需求,有效解决了当前农作物病虫害测报信息化中存在的问题,提高了植保人员的工作效率。

各级植保工作人员通过电脑或移动设备上的浏览器即可使用该系统,其优势在于:①无纸化办公,报表的传送和审批更为快捷高效;②规范的表格样式、标准的数据格式及可视化显示;③平台的转型和升级,更优秀的技术会提供更强大的功能和更安全的数据保护。利用可视化图表能帮助管理者和决策者提取直观有效的信息,利于科学防治决策的及时下达。同时在未来得到充分的病虫害数据后,会扩展数据挖掘模块来更好的帮助进行防治指导。

该系统在未来会根据业务需要和实际数据量的增长进行以下功能的扩展:在得到足量的实际应用数据后,建立机器学习模型来挖掘实际作物数据中潜在的价值,有利于对突发性、流行性病虫害的发生规律进行总结,分析预测作物病虫害的影响因素、发生趋势,从而实现及时下达防治决策和科学耕作指导的目的。

参考文献:

- [1] 中华人民共和国农业农村部.农业部出台加快推进现代植物保护体系建设意见[J].今日农药,2013(6):6-8.
- [2] 刘万才,黄冲.我国农作物现代病虫害测报建设进展[J].植物保护,2018,44(5):159-167.
- [3] 刘宇,刘万才,韩梅.农作物重大病虫害数字化监测预警系统建设进展[J].中国植保导刊,2011,31(2):33-35.
- [4] 刘万才,刘宇,龚一飞.论重大有害生物数字化监测预警建设的长期任务[J].中国植保导刊,2011,31(1):25-29.
- [5] 刘宇,刘万才,王学锋.水稻重大病虫害数字化监测预警平台的设计与实现[J].中国植保导刊,2009,29(12):5-9.
- [6] 刘万才,刘宇,曾娟,等.推进农业有害生物数字化监测预警建设刍议[J].中国植保导刊,2009,29(10):11-15.

- [7] 陈继光,宋显东,王春荣,等.黑龙江农作物病虫害在线监测管理系统开发与应用[J].中国植保导刊,2017,37(8):24-30.
- [8] 曹烁,王睿文,王鹏,等.河北省植物保护网络服务平台的实践与思考[J].中国植保导刊,2016,36(10):78-80.
- [9] 勾建军,刘莉,李秀芹,等.河北省监测预警体系建设面临的问题与工作重点思考[J].中国植保导刊,2018,38(7):84-87.
- [10] 郑庆伟.新疆农业有害生物远程预警防控指挥中心正式启动[J].农药市场信息,2016(5):61.
- [11] 张国鸣,罗川林,韩曙光.浙江省农作物有害生物监测预警体系建设的实践与思考[J].中国农技推广,2018,34(9):3-6.
- [12] 万宣伍,封传红,王胜,等.四川省农作物病虫害监测预警发展历程回顾及展望[J].中国植保导刊,2018,38(4):70-74+80.
- [13] 魏圆圆,王雪,王儒敬,等.基于 WebGIS 的农场生产管理信息系统的设计与实现[J].农业工程学报,2018,34(16):139-147.
- [14] 陈海中,张友华,刘家成,等.安徽省农作物病虫害监测预警平台的研制[J].中国植保导刊,2013,33(11):54-58.
- [15] 张建华,孔繁涛,吴建寨,等.农业物联网技术发展趋势预测[J].农业展望,2018,14(9):68-72.
- [16] 李棋.基于广域网的农业物联网监测系统的设计与实现[D].马鞍山:安徽工业大学,2017.
- [17] 杜克明,褚金翔,孙忠富,等.WebGIS 在农业环境物联网监测系统中的应用[J].农业工程学报,2016,32(04):171-178.
- [18] 白云.基于北斗导航与 WebGIS 的矿场人员监控系统详细设计[J].现代信息科技,2018,2(10):23-25.
- [19] 姜海燕,茅金辉,胥晓明,等.基于面向服务架构和 WebGIS 的小麦生产管理支持系统[J].农业工程学报,2012,28(08):159-166.
- [20] 赵聪.可视化库 D3.js 的应用研究[J].信息技术与信息化,2015(2):107-109.
- [21] 权鑫.基于 D3.js 的数据可视化系统框架设计与实现[D].北京:北京交通大学,2016.
- [22] 范伟梅.基于 D3 的数据可视化图表系统[D].广州:华南理工大学,2017.
- [23] 丁东辉.基于 Hadoop 和 D3.JS 的互联网+博物馆可视化平台的研究与实现[D].上海:东华大学,2017.
- [24] 蒋豪博,刘志平,张恒,等.基于 Qt 与 D3.js 的新型交互式图表开发[J].淮海工学院学报(自然科学版),2017,26(2):21-25.
- [25] 刘云浩,李沛.基于 Spring Boot 的后台服务器开发[J].中国科技信息,2018(17):61-62.
- [26] 张峰.应用 SpringBoot 改变 web 应用开发模式[J].科技创新与应用,2017(23):193-194.
- [27] 马雄.基于微服务架构的系统设计与开发[D].南京:南京邮电大学,2017.
- [28] 周永圣,侯峰裕,孙雯,等.基于 SpringCloud 微服务架构的进销存管理系统的设计与实现[J].工业控制计算机,2018,31(11):129-130+133.
- [29] 姜建武,李景文,陆妍玲,等.基于 RESTful API 的智慧旅游系统设计与实现[J].测绘与空间地理信息,2017,40(7):57-61.
- [30] KOBUSIŃSKA A, HSU C. Towards increasing reliability of clouds environments with RESTful web services[J]. Futur Gener Comput Syst,2018,87:502-513.
- [31] 张志,胡志勇.RESTful 架构在 Web Service 中的应用[J].自动化技术与应用,2018,37(10):33-37.