

SUSCEPTIBILITY DATA ANALYSIS AND VISUALIZATION FOR SURVEILLANCE IN ANTIMICROBIAL RESISTANCE

JIN, YUJIA
SUPERVISOR: DR. B.A. HERNANDEZ PEREZ

A Thesis submitted in fulfilment of requirements for the degree of
Master of Science
Applied Machine Learning
of Imperial College London

Department of Electrical and Electronic Engineering
Imperial College London
September 8, 2022

Abstract

In the early 20th century, the discovery of various antimicrobials solved the medical problem of death caused by infection with pathogenic microorganisms. With the large-scale use of antimicrobials, hospitals have found that during the use of antimicrobials, some microbes gradually develop resistance, making the antimicrobials ineffective. Each year, the number of resistant pathogenic microorganisms is increasing rapidly, but the number of new antimicrobials is not enough. If unchecked, address and deal with the Antimicrobial Resistance (AMR) problem, the annual number of deaths due to AMR will increase to 10 million by 2050. One of the most important causes of the AMR problem is the inappropriate and excessive use of antimicrobials by doctors and pharmacists. Therefore, this project builds an AMR surveillance web page system that can interact with users. The system provides doctors with clear and accurate AMR information and recommendations based on the collection, processing, analysis, and visualization of recorded susceptibility test data.

This project first cleaned and preprocessed the susceptibility test records downloaded from the Imperial College Healthcare NHS Trust. First, remove the data with obvious errors or inconsistencies in the original data. Second, respectively to extract corresponding analysis data from microorganism and antimicrobial two aspects and get two new processed CSV files. Third, Single Antimicrobial Resistance Index (SARI) and Antimicrobial Spectrum of Activity Index (ASAI) are used to analyze and process the data. At the same time, independent time intervals (ITI) and overlapping time intervals (OTI) are used for visualization to obtain a series of clinical data and maps that can provide reference. This project provides information for three levels: Pair Level, Antibiotic Level and Overall Level. Pair Level contains the number of total records per interval for a

pair, number of sensitive/resistant/intermediate records for a pair and multiple resistance time-series for a pair. Antibiotic Level contains a lateral bar chart of ASAI. Overall Level contains the overall Treemap.

In terms of designing and building the interactive AMR surveillance web system, the development environment of this project is mainly composed of Python3.7 +MongoDB+ Node.js + EChart + HTML + CSS + Javascript + VUE. This project carries out data cleaning, data validation and data analysis for susceptibility test record data from three dimensions (pari level,antibiotic level, overall level) based on Python. Python pyplot is used to verify the visual results of selected data. Then mongodb non-relational database is used for data storage and Node.js is used to build Node Server to provide restful API. Finally, VUE + HTML + CSS +javascript is used to develop the website, and the analysis results are displayed on the website using the Web visualization gallery Echart, and the interactive user interface is provided.

Acknowledgment

First of all, I would like to thank my family for their selfless love and great support.

Secondly, I would like to thank my supervisor Dr. B.A. Hernandez Perez For his patient guidance and meticulous help.

Finally, I would like to thank all the teachers at Imperial for the learning environment and continued encouragement.

With the help of my family, my supervisor and teachers at Imperial College London, I completed this project.

Contents

Abstract	3
Acknowledgment	5
Contents	7
List of Figures	11
List of Tables	13
Abbreviations	15
Chapter 1. Introduction	17
1.1 Background	17
1.2 Purpose and significance	18
1.3 Status of international research	19
1.4 Main work of the project	21
1.5 Overall structure of the report	22
Chapter 2. Related technology and theoretical basis	23
2.1 Basic theory	23
2.1.1 Data collection, cleaning and preprocessing	23
2.1.2 AMR data definition and data processing	24
2.1.3 AMR data analysis and visualization	27
2.2 Related technologies	28
2.2.1 Python data development and data processing	28
2.2.2 Plotly visualization framework	28
2.2.3 MongoDB database development tool	29
2.2.4 Nodejs-Based Web Server Development	29
2.2.5 ECharts Visualization Framework	29
2.2.6 AdminLTE website template	30

Chapter 3. System requirement analysis	31
3.1 Data requirement analysis	31
3.1.1 Data Indicator Requirement Analysis	31
3.1.2 Data visualization requirement analysis	31
3.2 The website requirement analysis	32
Chapter 4. System design	35
4.1 Design of system architecture	35
4.2 Design of function modules	36
4.2.1 Design of data collection module	36
4.2.2 Design of data processing module	36
4.2.3 Design of data analysis module	37
4.2.4 Design of data visualization module	38
4.2.5 Design of the website	40
4.3 Introduction to the Development Environment.	43
Chapter 5. System implementation	45
5.1 Overall system implementation scheme	45
5.2 Data processing module implementation	46
5.3 Data analysis module implementation	47
5.3.1 Pair Level	47
5.3.2 Antibiotic Level	47
5.3.3 Overall Leve	48
5.4 Data visualization module implementation	48
5.4.1 Data visualization based on Python Plot	48
5.4.2 Data visualization based on Echarts Visualization	50
5.5 The website Implementation	52
5.5.1 Database implementation and table design	52
5.5.2 Back-end service implementation	54
5.5.3 Front-end implementation	59
Chapter 6. System test	63
6.1 Logging in to a module	63
6.1.1 Test Purpose	63
6.1.2 Experimental Environment and Configuration	63
6.1.3 Test Procedure and Result	63
6.2 Pair Level function test	65
6.2.1 Test Purpose	65
6.2.2 Experimental Environment and Configuration	65

6.2.3	Test results and analysis	65
6.3	Antibiotic Level function test	67
6.3.1	Test Purpose	67
6.3.2	Experimental Environment and Configuration	67
6.3.3	Test results and analysis	67
6.4	Overall Level functional test	68
6.4.1	Test Purpose	68
6.4.2	Experimental Environment and Configuration	68
6.4.3	Test results and analysis	68
Chapter 7.	Discussion and conclusions	71
7.1	Disscusion	71
7.2	Conclusions	72
Bibliography		73

List of Figures

4.1	System architecture design	35
4.2	Code structure	37
4.3	Dataframe	37
4.4	SARI ITI time-series for a pair	38
4.5	SARI OTI time-series for a pair	39
4.6	Multiple SARI OTI and ITI time-series for a pair	39
4.7	ASAI for a specimen type	40
4.8	Overall Treemap	40
4.9	Examples of figures in Echarts	41
4.10	Product Requirement Document for pair level	41
4.11	Product Requirement Document for antibiotic level	42
4.12	Product Requirement Document for overall level	42
5.1	Three layers of the overall system implementation	45
5.2	The mapping relationship in the file	46
5.3	The visualization code in python plot	49
5.4	The visualization of SARI time-series in a pair	50
5.5	Examples of graphs in Echarts	50
5.6	The specific visualization code in Echarts	51
5.7	The visualization of Echarts example	51
5.8	The table structure in MongoDB	52
5.9	Initialize the NodeJS project package	55
5.10	Specify the HttpServer listening port and API route	55
5.11	Run NPM run start-server to start the service	56
5.12	'Hello World' Test result	56
5.13	Query the result in MongoDB	57
5.14	Search for the rows that match the conditions in the data_susceptibility table	58
5.15	Test result	58
5.16	Webpage for pair level	60

5.17 Webpage for antibiotic level	61
5.18 Webpage for overall level	61
5.19 Login page	62
5.20 User profile page	62
6.1 Test for incorrect user name	64
6.2 Test for incorrect user name	64
6.3 Test for correct name and password and successfully login	65
6.4 Three drop-down lists	66
6.5 Test for pair level	66
6.6 Test for antibiotic level	67
6.7 Test for overall level	68
6.8 Test for hover display floating box	69

List of Tables

Abbreviations

AMR: Antimicrobial Resistance

SARI: Single Antimicrobial Resistance Index

ASAI: Antimicrobial Spectrum of Activity Index

ITI: Independent Time Intervals

OTI: Overlapping Time Intervals

Chapter 1

Introduction

1.1 Background

In the late 19th century, patients were often killed by infections caused by pathogenic microorganisms. Infection by pathogenic microorganisms is one of the leading causes of death worldwide. From the beginning of the 20th century to the middle of the 20th century, with the discovery, development and large-scale use of antimicrobials, the problem of death caused by pathogenic microorganisms infection was largely solved.

However, with the large-scale use of antimicrobials and the development of medical technology, hospitals have found that in the process of using antimicrobials, some microorganisms will gradually develop resistance, which makes antimicrobials ineffective. This can greatly reduce the effectiveness of the treatment. What's more, the resistant organism can also be passed from patient to patient. Every year, the number of resistant pathogenic microorganisms increases rapidly, but the number of new antimicrobials is not enough [1]. After the golden age of the rapid development of antimicrobials has passed, with the enhancement of the resistance of pathogenic microorganisms, common injuries, pathogenic microbial infection and transmission may again become large medical problems to be overcome by human beings [2].

According to the World Health Organization (WHO), nearly half a million new MDR-TB patients are detected each year alone. Among these, about 34.1% of patients

died. Across the full range of AMR data, about 700, 000 people die each year from antimicrobial resistance. If the problem of AMR is not controlled, solved and dealt with, the annual number of deaths due to AMR will increase to 10 million in 2050 [3]. Due to the rapid development of the AMR problem and the major threat to global human health, hospitals need to establish a suitable AMR monitoring system. By means of data collection, processing, analysis and visualization, hospitals can deal with the increasingly serious problem of AMR, inhibit the spread of resistant pathogenic microorganisms and save more patients' lives.

1.2 Purpose and significance

One of the most important causes of the AMR problem is the inappropriate and excessive use of antimicrobials by doctors and pharmacists. Among them, over-the-counter antimicrobials are overused in many countries, especially in underdeveloped and developing countries [4]. Due to the lack of sufficient clinical data, information and knowledge, doctors are often unable to quickly and correctly choose the most appropriate antimicrobials. Instead, they choose antimicrobials that are broadly applicable and can treat more pathogenic microorganisms. This further leads to the emergence and proliferation of AMR problems. In fact, with enough clinical data and information, it is difficult for doctors to quickly find out the trend of antimicrobial resistance, the range of use, the characteristics of antimicrobials, the recommended dosage of antimicrobials and other key information. This requires further analysis and processing of these clinical data, such as data mining and visualization, so that doctors can quickly get the comprehensive and accurate information and guidance and suggestions they need.

Common data provided in the sensitivity test record are sample date, laboratory number, sample type, antimicrobial name, microorganism name, and sensitivity test results (sensitive/intermediate/resistant). By processing, analyzing and visualizing these data and information, doctors can find information such as number of total records per interval for a pair, the resistance time-series for a pair, the application scope and characteristics of antimicrobials, and the overall treemap. If more kinds of experimental data are

collected, such as the patient's previous underlying disease, the current regional incidence, and so on, through the processing of some AMR data indicators, statistical analysis methods and data mining, doctors can even find out the biological reasons for the occurrence of AMR, the incidence of the local population and the transmission route of AMR [5]. With more comprehensive, accurate and real-time information, and analysis using appropriate data-mining methods, doctors can quickly and accurately find the right antimicrobials and even get the right dosage recommendation. All of these AMR data, information and visualizations, provide doctors with recommendations and references for clinical treatment. Doctors can refer to the specific local situation and the specific situation of the patient, rather than enforcing the systematic recommended treatment [6]. In the developed monitoring system and web page, it is also very necessary to update the sensitivity test record data and information in time [7].

The purpose of this project research is to contribute to solving susceptibility data analysis and visualization for surveillance in Antimicrobial Resistance. To solve this problem, it is necessary to process, analyze and visualize the sensitivity test record data. At the same time, in order to facilitate doctors to quickly access and get the information and guidance that they need at the moment, it is also necessary to build an interactive web page containing the recorded data of various sensitivity tests. In this web page, doctors can get SARI, ASAI, overall treemap and other information and advice to help doctors make rational decisions.

It is of great significance to collect and analyze the sensitivity test record data for improving the treatment of patients in time, saving the lives of patients and reducing the transmission of resistant microorganisms.

1.3 Status of international research

Based on the increasingly serious problem of AMR and its serious consequences, countries and localities are trying to find new ways to solve the problem and establish new systems to solve the problem. WHO, Europe and the United States respectively have

established AMR monitoring methods and systems. For example, WHO established the Global Antimicrobial Resistance Surveillance System (GLASS). Among them, the United States attaches great importance to the research of AMR, and has invested huge national resources to study the reasons for the emergence and rapid development of AMR, and to curb and solve the problem of AMR. In recent years, the United States has been focusing on the analysis, design and creation of new AMR monitoring methods and systems [8].

There are two common approaches in data and information collection. The first is to export directly from the medical system and then use an offline approach for data mining and subsequent processing. The second method is to integrate the data collection module directly into the AMR monitoring system. Real-time collection of experimental data through the whole system, such as equipment information, sample type, death risk, complication information, etc. After that, data processing and visual analysis are carried out directly in the system in the form of online and real-time upload.

In the process of data mining and processing, the simplest is to apply simple statistical methods. The information and data obtained by simple statistical methods are very limited, and it is difficult for doctors to get effective guidance and suggestions [9]. Therefore, in recent years, more methods have been adopted to carry out data mining and knowledge discovery, such as machine learning and expert systems, which are applied to different analysis perspectives. For example, clustering is usually used to identify abnormal records in sensitivity test record data [10].

In the process of analyzing and visualizing the recorded data of sensitivity tests, the most commonly used method in the world is heat map [11] [12]. In the design and establishment of the AMR monitoring system, different scholars have designed a variety of dashboards to meet the actual needs of different regions and doctors. For example, Stedtfeld designed a dashboard [8]. It can analyze the data of sensitivity test records from the perspective of microbiological factors, analyze the genes of microorganisms that produce resistance, and get some possible factors and prevention methods that lead to the resistance of microorganisms in clinical practice.

1.4 Main work of the project

On the basis of sufficient demand analysis and in-depth research, the main work of this project is to design and implement an AMR monitoring web page system that can interact with users. Based on the collection, processing, analysis, and visualization of sensitivity test records, the system provides doctors with the clear and accurate AMR critical information and clinical recommendations they need.

The specific approach is to use Python to clean, process and analyze the sensitivity test record data. Then, according to doctors' requirements, use Plotly to visualize key data such as time series bar chart, line chart, overall treemap, etc. Finally, through Python3.7 +mongodb+ Node.js + Echart + HTML + CSS + Javascript + VUE to build a web page system that allows doctors to conduct interactive operations. The webpage system contains three levels of visualization, pair level, antibiotic level and overall level. Pair Level contains the number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair and multiple resistance time-series for a pair. Antibiotic Level contains a lateral bar chart of ASAI. Overall Level contains the overall Treemap. Doctors can interact, select, and manipulate on demand, getting information that helps them make quick judgments and treatment decisions.

The main research contents include:

- 1) Study the current international mainstream solutions to the AMR problem. Study the current calculation methods and shortcomings of common AMR indicators.
- 2) Study the collection, processing, analysis and visualization of current AMR common data.
- 3) Study the current development technology and platform of the AMR monitoring system. Determine the development technology, the development environment and the main functions of the system that will ultimately be used in the project.
- 4) Design the system interface, define functions and study the technical framework.
- 5) Develop the system and realize the function.
- 6) Test, repair and maintain the system.

1.5 Overall structure of the report

This paper is divided into 7 chapters, the overall structure of the paper is as follows:

- 1) Chapter One Introduction. This chapter mainly introduces the background, purpose and significance of the paper, and the research status of AMR monitoring. At the end of this chapter, the main research content of this project and the overall structure of the paper are explained.
- 2) Chapter Two Related technology and theoretical basis. This chapter first introduces some basic theoretical knowledge about AMR. Then it introduces the techniques used in the development of the AMR monitoring system, such as Python, MongoDB, Node.js, ECharts, etc.
- 3) Chapter three System requirement analysis. This chapter carries on the detailed requirement analysis to the thesis research system, including the data requirement analysis and the website requirement analysis.
- 4) Chapter four System design. This chapter introduces the design of the system and explains the development environment of the system. And the system architecture design to the system of each module function design are elaborated in detail.
- 5) Chapter five System implementation. This chapter has carried on the detailed elaboration to the system of each function module and the website final realization.
- 6) Chapter Six System test. In this chapter, the functional test of the system is explained in detail, and the visual results of system data analysis are partially displayed.
- 7) Chapter Seven Discussion and conclusions. This chapter first discusses and summarizes the content of the paper, and then explains the problems existing in the current system and the future improvement of the system.

Chapter 2

Related technology and theoretical basis

2.1 Basic theory

2.1.1 Data collection, cleaning and preprocessing

Susceptibility test record data susceptibility-2020. csv in this project is downloaded from the Imperial College Healthcare NHS Trust. Data actually contains mainly sample date, laboratory number, specimen code, specimen name, specimen description, microorganism code, microorganism name, antimicrobial code, antimicrobial name, sensitivity outcome and other information. The widely accepted method in clinical practice is to first classify the samples according to their specimen type. After that, according to the requirements of different data indicators, a microorganism and an antimicrobial are classified as a group.

Data processing mainly includes data cleaning and data preprocessing. Data cleaning is to remove the obvious errors or inconsistent data in the original data, so as to provide more accurate data for subsequent data processing. For example, the field is missing (date received is missing, specimen is missing). The data preprocessing for this project is based on the source file susceptibility-2020. CSV. Through dimension calculation, this article respectively originates from two dimensions (antimicrobial and microorganism) to extract the corresponding analysis data and get two processed CSV files. The two new processed

files are microorganism.csv and antimicrobial.csv.

2.1.2 AMR data definition and data processing

In order to quantify the study of antimicrobial resistance, some AMR data indicators have been defined in both clinical and theoretical studies. These indicators include, but are not limited to, Single Antimicrobial Resistance Index (SARI), Single Antimicrobial Resistance Trend (SART), antibiotic spectrum index (ASI), Antimicrobial Spectrum of Activity Index (ASAI) [13], Multiple Antimicrobial Resistance Index (MARI) [1], Drug Resistance Index (DRI). Among them, SARI, SART and MARI are commonly used and widely accepted indicators. In order to visually and clearly show the situation of antimicrobial resistance to doctors, SARI and ASAI are selected as two indicators in this project.

The definition of SARI

Sari is the degree of resistance of a particular microorganism to a particular antimicrobial in the same sample type. In the sensitivity test record, each set of data is given a result, sensitive, intermediate or resistant. In actual data processing, Sari is calculated as Equation 2.2. The value of SARI may vary slightly between reports, depending on whether I is taken into account. In the following formula, S stands for sensitive, I for intermediate, and R for resistance.

$$SARI = \frac{R + I}{R + I + S} \quad (2.1)$$

With SARI, doctors can know exactly in a particular sample type, how resistant a particular pair of microorganism and antimicrobial is. Thus, the follow-up treatment plan can be adjusted accordingly.

The definition of ITI and OTI

To help doctors better analyze AMR data, this project plotted the change of Sari value over time. In the traditional method, the time sequence diagram is usually drawn at inde-

pendent time intervals. This is a way to compute data in periods that do not overlap each other. For example, a time sequence diagram is drawn in the unit of each month, that is, SARI is calculated with the sample data of each month. This method has obvious limitations, and two concepts of accuracy and granularity [14] are introduced here. Accuracy refers to how accurate Sari is. The more samples the system uses to calculate Sari, the more accurate Siri's numbers will be. The more times the Sari value is computed over the same length of time, the greater the granularity. If the time unit is monthly, the accuracy is lower. Because the number of samples in a month is limited. Taking year as the time unit, the granularity is lower. Because Sari is only calculated once a year.

To solve this problem, this project introduced overlapping time intervals [15]. There are two parameters in the overlapping time interval. First, specify a frequency of calculating Sari values using sensitive data records, like a month. Second, specify a compute period longer than the test frequency period, such as 3 months. It is equivalent to calculating the common susceptibility data of the current month and the previous two months once a month. In this way, the SARI calculated for each month contains three months of sensitivity data records and SARI has a high accuracy. At the same time, SARI is calculated 12 times in a year, which also has a large granularity. This project uses the following notation to indicate the different time intervals. This project uses the following notation $30D_1$, $30D_1$, D_{90D} , $30D_{180D}$ to indicate the different time intervals.

$30D_1$ is the independent time interval.

$30D$ means frequency is 30 days, that is, SARI is calculated once a month.

The lower case 1 means that the sample calculated each month is from the previous month.

$90D_1$ is also an independent interval.

$90D$ represents the frequency of 90 days, that is, SARI is calculated every three months.

The lower case 1 means that the sample calculated each 3 month is from the previous 3 months.

$30D_{90D}$ is the overlapping time interval.

$30D$ means frequency is 30 days, that is, SARI is calculated once a month.

The lower case $90D$ means the sample calculated each month is from the previous 3 months.

$30D_{180D}$ is the overlapping time interval.

30D means frequency is 30 days, that is, SARI is calculated once a month.

The lower case 180D means the sample calculated each month is from the previous 6 months.

With the OTI calculation method, physicians can obtain high accuracy and granularity at the same time, and have the flexibility to observe different SARI values according to the requirements.

The definition of ASAII

In 2017, ASI [16] was defined as a new indicator of AMR. ASI shows different antimicrobials are effective on which range of microorganisms, but only includes some specific microorganisms. In order to include a wider range of microorganisms and calculate more accurate values, a new improved index, ASAII [15], was adopted in this project. ASAII shows different antimicrobials are effective on which The range of microorganisms.

Antibiotics are a class of antimicrobials that act only on bacteria. In this project we are talking about the activity spectrum of antibiotics. Bacteria can be divided into Gram-negative bacteria and Gram-positive bacteria, which can be quickly distinguished by 3% potassium hydroxide solution [17]. In the biological classification, biology can be generally divided into domain, phylum, class, order, family, genus and species. SARI is an indicator in the species dimension, while ASAII is an indicator in the genus dimension. The specific calculation is shown Equation 2.2 ,

Among them, N_{gn} represents the total number of genus in microorganisms, w_{gn} represents the weight of the genus, N_{sp}^{gn} represents the total number of microorganisms' species in a certain genus, w_{gn} represents the weight of species under a certain genus, and $SARI_{sp}^{gn}$ represents the SARI value of microorganisms of a certain species under a certain genus, th represents the threshold of SARI.

$$ASAII_{th} = \frac{1}{N_{gn}} \sum_{\forall gn} w_{gn} \cdot \left(\frac{1}{N_{sp}^{gn}} \sum_{\forall spp^{gn}} w_{gn}^{sp} \cdot \text{effective } th(SARI_{sp}^{gn}) \right) \quad (2.2)$$

where

$$\sum_{\forall gn} w_{gn} = 1 \quad \sum_{\forall sp} w_{gn}^{sp} = 1 \quad \text{effective } th(\text{SARI}) = \begin{cases} 1 & \text{if SARI} \leq th \\ 0 & \text{otherwise} \end{cases}$$

ASAI was calculated under the dimension of genus, using SARI weighted values of species dimension. In the case of Gram-negative bacteria, to calculate ASAI, firstly select samples of the same type. Secondly, the calculation method of the ASAI value of a certain antibiotic is the weighted average of the sari of all microorganisms in all genus dimensions under a certain antibiotic. Among them, the calculation method of the sari value of microorganisms in a genus dimension, is the weighted average (weighted by the number of samples) of the effective sari of microorganisms in all species dimensions in this genus. When the SARI value is less than the specified value, it is considered that the antibiotic has a therapeutic effect on the bacteria, that is, the effective SARI value mentioned above. The antibacterial spectrum helps clearly find the characteristics of each antimicrobial doctor.

2.1.3 AMR data analysis and visualization

This project will visualize the obtained index data according to the different needs of clinicians. In general, the intuitive information required by the clinician is the SARI value, number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair, the resistance time-series for a pair, etc. Visualization can be expressed by pie charts, line charts, bar charts and other geometric figures. It can also be expressed through color difference, relative position, area size and other information [18]. After weighing and comparing the needs of the physician and the pros and cons of various visualization methods, this project included three levels of data visualization: Pair Level, antibiotic level and Overall level.

1) Pair Level. After selecting the sample type, antimicrobial and microorganism, Pair Level can display number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair, the resistance time-series for a pair, and the

time series line chart of ITI and OTI.

- 2) antibiotic level. If a specimen type was chosen, the antibiotic showed horizontal bars of ASAII against Gram-negative and Gram-positive bacteria, respectively.
- 3) Overall level. Overall Level Displays the overall Treemap image. The doctor can get the following information by means of hovering, antimicrobial code, the number of samplesmicroorganism code specimen code microorganism name, antimicrobial name, sari hard, sari medium and sari soft.

2.2 Related technologies

2.2.1 Python data development and data processing

Python is not only a high-level programming language, but also a high-level scripting language that combines interpretation, compilation, interactivity and object-oriented programming, so Python also includes all the features of scripting languages. At the same time, it is also a mature and stable computer general-purpose development language.

Compared with low-level languages, the design has strong readability and clear syntax structure. It takes less time to write Python programs, the program is shorter, and it is easier to ensure the correctness of the program. Therefore, it is also called executable pseudocode. As a high-level language, Python is easy to learn, easy to read, easy to maintain, extensible, embeddable, and portable. The same program can run on different types of systems without modification. Running on the system, you can also embed Python programs into C/C++ programs.

2.2.2 Plotly visualization framework

Plotly is an open source tool for writing, editing and sharing interactive data visualizations. It presents information by constructing interactive charts in web form based on browser display. Plotly can create up to dozens of different charts and maps such as bar, scatter, pie, histogram, and more. And Plotly supports online editing. It is a drawing system that integrates multiple drawing methods of online drawing through menu operation and offline

drawing through code. It is also very easy to import and export data, you can directly upload CSV files or connect to SQL databases to create charts, and it has multiple API interfaces, such as matlab, R, and Excel. find a figure later

2.2.3 MongoDB database development tool

MongoDB is a popular NoSQL database product. It is an open source free, easy to expand, free table structure (schema free), high-performance and document-oriented non-relational database. The MongoDB database was developed by the 10gen team in October 2007. It was developed and first released in February 2009, and the MongoDB database developed rapidly. The MongoDB database structure is that the database contains collections, the collections contain documents, and the documents contain one or more key-value pairs. Compared with the traditional relational database, the collection in MongoDB is similar to the table in the relational database. The difference is that the collection does not have a fixed header. The document in the MongoDB collection is similar to a row of data in the relational database. The difference is that the document is not fixed. model. The specific differences are shown in the table.

2.2.4 Nodejs-Based Web Server Development

Node.js is a JavaScript runtime environment based on Google browser's V8 engine, which can be used to build high-speed, scalable network programs. The Node.js framework enables JavaScript to be developed in the same way as Java, Python, PHP and other development languages, that is, the server side. Simply put, Node.js is JavaScript that runs on the server. So using Node.js, you can use the JavaScript language to develop all the code on the server side and all the code on the client side at the same time, which is convenient and fast.

2.2.5 ECharts Visualization Framework

The full name of ECharts is Enterprise Charts. It is a data visualization tool under Baidu. It is an open source, web-based cross-platform framework. It is not only compatible with

many different types of browsers, but also supports the rapid construction of interactive visual charts. ECharts not only has a high-performance graphics renderer, but also the core based on the declarative visual design language makes it possible to define built-in chart types by itself, and has high scalability .

As a Javascript chart library, ECharts not only has many corresponding templates, but also various types of graphs and charts, such as area charts, bubble charts, column charts, pie charts, scatter charts, maps, candlestick charts, event river charts and funnels Figure etc. At the same time, ECharts also has a rich API, with seven different interactive components, and supports a variety of charts for mashup display, with very good visual effects. And ECharts can also use Ajax technology and its own event mechanism to link the data theme graphical interface in the foreground with the data in the background. When the data changes or new data is imported, the dynamic display effect of the data will change accordingly. On the basis of enhancing the user experience, the ability to integrate and mine data has been improved.

2.2.6 AdminLTE website template

AdminLTE is a front-end framework, it is an open source template theme tool based on Bootstrap3 and JQuery, and based on Bootstrap3, it designs background management panels and dashboard templates suitable for its own style. AdminLTE can not only use most styles of Bootstrap3, but also provides a very practical style package and a series of responsive components, and has built-in multiple template pages, such as login template, registration template, personal template, 404 template, 500 template and Blank templates, etc., these can be viewed in the Examples menu after downloading. There are not only some EasyUI components, but also some custom CSS styles. The box style is the most basic and most commonly used among the custom CSS styles. In addition to Bootstrap's own modal window, navigation, and drop-down menu, the JS plugins in it integrate some well-known third-party plugins, such as Select2, CK Editor, ICheck, DataTables, and so on.

Chapter 3

System requirement analysis

3.1 Data requirement analysis

3.1.1 Data Indicator Requirement Analysis

In this project, the project mainly calculated two kinds of AMR data indicators – SARI and ASAII through a series of steps such as data cleaning, preprocessing, processing and analysis. SARI is an indicator often used in clinical practice, which allows doctors to clearly know the resistance degree of a specific microorganism to a specific antimicrobial in a specific sample type in a specific period of time. ASAII shows marketers antimicrobials are effective on which range of microorganisms, in different types of samples. The antimicrobial spectrum of activity helps doctors clearly find the characteristics of each antimicrobial. In clinical treatment, according to SARI and ASAII, doctors can choose the appropriate treatment plan more accurately and quickly.

3.1.2 Data visualization requirement analysis

After obtaining specific data indicators, in order to facilitate doctors to understand the required AMR information more clearly and quickly, the project will visualize the data according to different needs.

- 1) The resistance time-series for a pair.

This project requires visualization of SARI values over time. The time series map of SARI

is displayed in the form of broken line charts. Doctors can quickly view the SARI value, number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair, the resistance time-series for a pair by selecting the sample type, microorganism and antimicrobial, clicking different parts of the broken line and other the projectb interactive ways. Among them, SARI is calculated by independant time interval and overlapping time interval respectively.

2) The spectrum of activity of antimicrobials

In this project, the project discuss the activity spectrum of antibiotics, and it is necessary to perform aggregation analysis on ASAI to obtain the horizontal bar chart. ASAI shows different antibiotics are effective on which range of Gram-negative and Granger-positive bacteria, in different sample types. As a result, doctors can choose antibiotics that meet a patient's needs, rather than a broad-spectrum antibiotic that works against more bacteria.

3) Treemap

In order to give the doctor a better overall view of the sensitivity test record data, key information such as antimicrobial name, microbe name, number of samples, and SARI value can be seen on a single graph. This project also visualizes the Treemap of the over-all information. By clicking on different parts of the Treemap on the the projectb page, the doctor can quickly view the SARI value, the total number of samples, the name of the microorganism, the name of the antimicrobials, and other information.

Therefore, the data visualization requirements of this project are mainly to show number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair, ITI and OTI multiple resistance time-series for a pair, Treemap word cloud charts and horizontal bar charts of ASAI.

3.2 The website requirement analysis

Based on the above data requirement analysis, the final requirement of this AMR surveillance system is to display the Python data results of the single pair that have been verified locally to users through the projectb data visualization. Build relevant the projectbsites, display the visual results (such as line chart, bar chart, word tree chart) on the the pro-

jectbsite, and provide a friendly user interaction interface, to realize the user such as pull-down selection of different pair level, hover view different dimensions of the distribution of interaction.

Chapter 4

System design

4.1 Design of system architecture

The current architecture design diagram is shown in the Fig. 4.1:

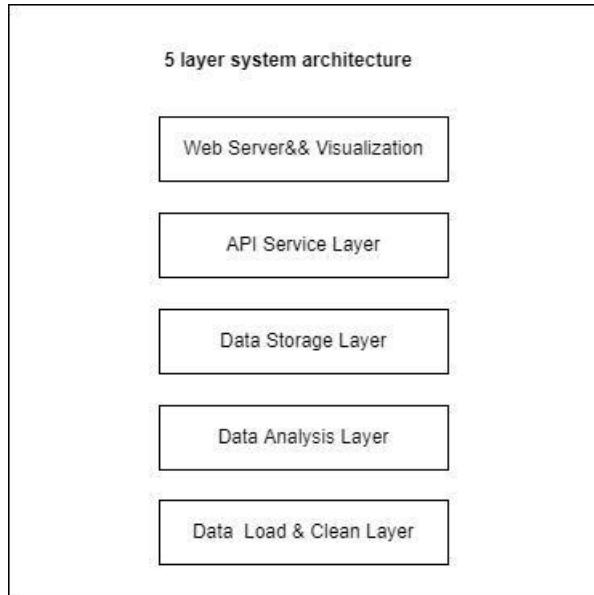


Figure 4.1: System architecture design

Architecture design diagram, is a multi-layer structure. The fifth layer is data collection and data preprocessing, providing three CSV files to the upper layer. The fourth layer is the data analysis layer. Through three CSV files, based on Python Pyamr library, the feasibility of various index data is analyzed file by file, and the data singleton test

and data visualization test are carried out. The third layer is the database layer, which stores the data and business data of the above two layers in the MongoDB database. The second layer is the server layer: it calls the database data of the third layer and provides restful api for the upper layer. The first layer is the user interaction layer: according to the selection of user interaction, the API of the server layer is called, and the data returned by the API and data visualization components Echarts are used to provide users with visual views.

4.2 Design of function modules

4.2.1 Design of data collection module

Susceptibility test record data susceptibility-2020. CSV for this project is downloaded from the Imperial College Healthcare NHS Trust. The data mainly contains specimen Name, specimen Name, Antimicrobial Name, sensitivity outcome and other information.

4.2.2 Design of data processing module

The data processing module is mainly based on source files to produce the actual data needed by subsequent modules (such as data analysis and data visualization). The data processing module mainly includes data cleaning and data preprocessing, which are as follows:

- 1) Data cleaning: Read the source file and remove the dirty data points that do not meet the requirements of this project, including but not limited to missing fields (such as missing date_received, missing specimen), obvious abnormal data index, expired data (too old), etc. The final result is relatively clean data. This is beneficial to provide subsequent data analysis accuracy.
- 2) Data preprocessing: Based on the source file susceptibility-2020. CSV, through dimension calculation, from the two dimensions of microorganism and antimicrobial, respectively, to extract the corresponding analysis data. Through the two scripts create_db_microorganism.py and create_db_antimicrobial.py, data preprocessing is realized,

and two processed csv files, microorganism.csv and antimicrobial.csv, are obtained. The details of the three files are shown in the Fig 4.2 and Fig. 4.3,

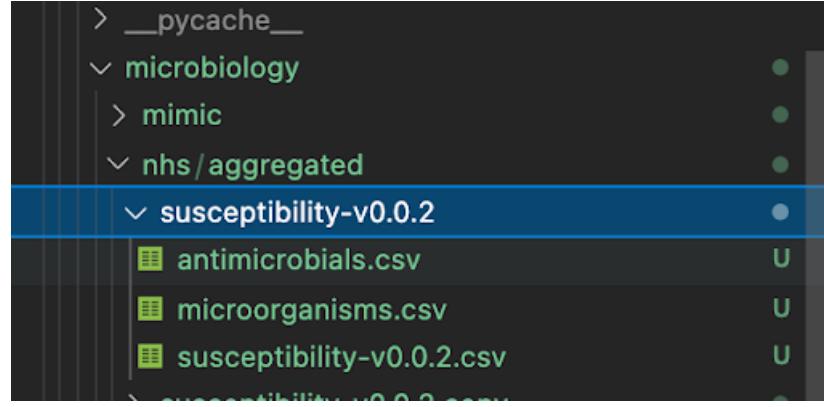


Figure 4.2: Code structure

```
# datasets get: [INFO] python ./online_timeseries_V11.py
/Users/david-dai/test/jinyujia/pyAMR/datasets/microbiology/nhs/aggregated/susceptibility-v0.0.2
/Users/david-dai/test/jinyujia/pyAMR/datasets/microbiology/nhs/aggregated/susceptibility-v0.0.2/antimicrobials.csv
/Users/david-dai/test/jinyujia/pyAMR/datasets/microbiology/nhs/aggregated/susceptibility-v0.0.2/microorganisms.csv

Data:
   date_received      date_outcome laboratory_number specimen_code      specimen_name ... antimicrobial_code antimicrobial_name sensitivit y mic reported
0  2020-01-05 08:44:00  2020-04-05 00:00:00  F1413221    URNCUL     Urine Micro ...      AME      amikacin sensitive NaN   N
1  2020-01-05 08:44:00  2020-04-05 00:00:00  F1413221    URNCUL     Urine Micro ...      ACAZ      ceftazidime sensitive NaN   Y
2  2020-01-05 08:44:00  2020-04-05 00:00:00  F1413221    URNCUL     Urine Micro ...      ACIP      ciprofloxacin sensitive NaN   Y
3  2020-01-05 08:44:00  2020-04-05 00:00:00  F1413221    URNCUL     Urine Micro ...      AGEN      gentamicin sensitive NaN   Y
4  2020-01-05 08:44:00  2020-04-05 00:00:00  F1413221    URNCUL     Urine Micro ...      AMER      meropenem sensitive NaN   N
...
47128 2020-12-08 23:11:00  2020-08-16 00:00:00  W1735492  MNGCUL  MRSA Nose and Groin Screen ...      ANEO      neomycin resistant NaN   N
47129 2020-12-08 23:11:00  2020-08-16 00:00:00  W1735492  MNGCUL  MRSA Nose and Groin Screen ...      APEN      penicillin resistant NaN   N
47130 2020-12-08 23:11:00  2020-08-16 00:00:00  W1735492  MNGCUL  MRSA Nose and Groin Screen ...      ARIF      rifampicin sensitive NaN   N
47131 2020-12-08 23:11:00  2020-08-16 00:00:00  W1735492  MNGCUL  MRSA Nose and Groin Screen ...      ATET      tetracycline sensitive NaN   N
47132 2020-12-08 23:11:00  2020-08-16 00:00:00  W1735492  MNGCUL  MRSA Nose and Groin Screen ...      ATRE      trimethoprim sensitive NaN   N

[47133 rows x 13 columns]

Columns:
Index(['date_received', 'date_outcome', 'laboratory_number', 'specimen_code', 'specimen_name', 'specimen_description', 'microorganism_code', 'microorganism_name', 'antimicrobial_code', 'antimicrobial_name', 'sensitivity', 'mic', 'reported'], dtype='object')
```

Figure 4.3: Dataframe

4.2.3 Design of data analysis module

Through the data processing module, the project will get 3 data files, susceptibility.csv, microorganism.csv and antimicrobial.csv. After importing the three files into the data path of the data module, the project can run the script for data analysis. The data analysis module mainly has the following two functions:

- 1) By calling various calculation methods in PyAMR's core method, the corresponding SARI/ASAI data indicators are calculated and the output data after data processing is

verified to be correct or not.

- 2) Verify that if the current visualization needs can be met by calling Python's visualization tools.

For this purpose, this module has two main parts.

- 1) The first part is by calling `pyamr.core.asai`, `pyamr.core.sari`, `pyamr.core.freq`. After importing data through `pyamr.datasets.load`, calculate ASAII, SARI, freq and other indicators respectively.
- 2) The second part is to verify the feasibility of data visualization by calling the PyPlot Python library.

4.2.4 Design of data visualization module

The data visualization module has two main parts:

- 1) The first part is the data visualization part based on Python Pylot library, which mainly displays the corresponding data analysis results from the perspective of Timeline, Barline, Bargraph, Treemap and other visualizations. The specific visualization results are as Fig 4.4, Fig 4.5, Fig 4.6, Fig 4.7 and Fig 4.8.

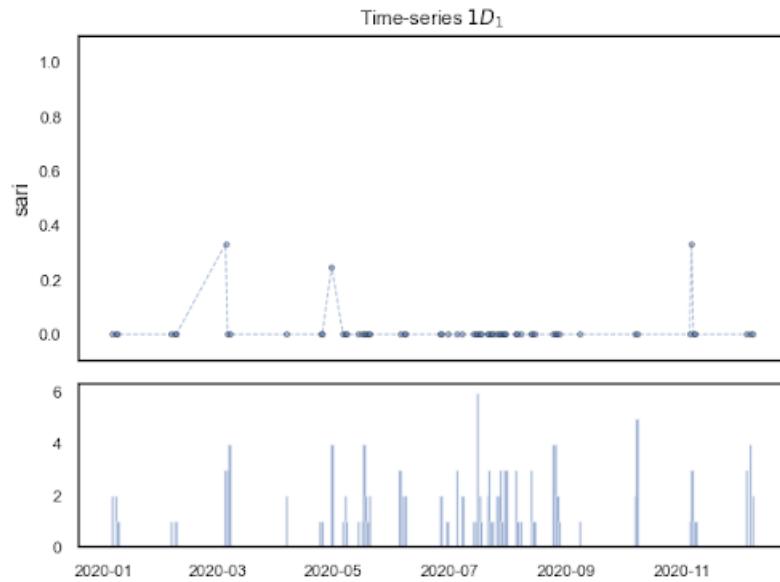


Figure 4.4: SARI ITI time-series for a pair

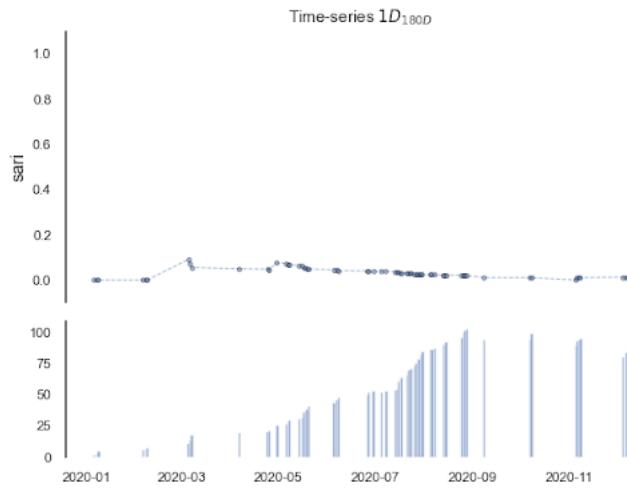


Figure 4.5: SARI OTI time-series for a pair

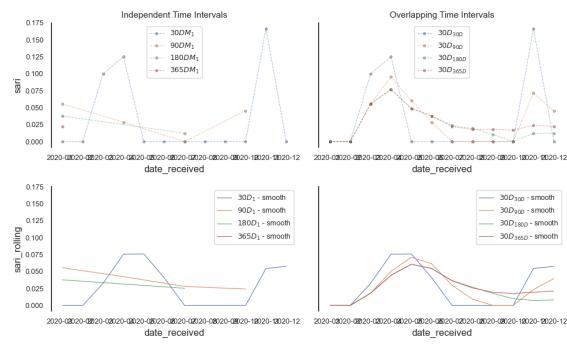


Figure 4.6: Multiple SARI OTI and ITI time-series for a pair

2) The second part is based on the web Echarts visual component library, which is mainly implemented from Timeline, Bargraph, Treemap and other perspectives as Fig 4.9.

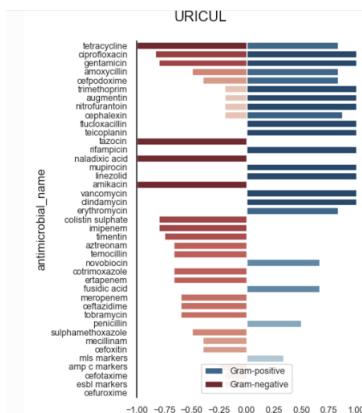


Figure 4.7: ASAII for a specimen type



Figure 4.8: Overall Treemap

4.2.5 Design of the website

According to the demand analysis, based on the above data analysis and data visualization requirements, the overall design of the website is as Fig 4.10, Fig 4.11 and Fig 4.12:



Figure 4.9: Examples of figures in Echarts



Figure 4.10: Product Requirement Document for pair level

The website pages include:

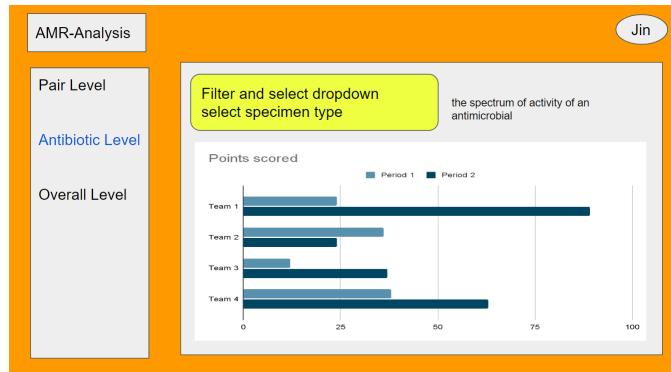


Figure 4.11: Product Requirement Document for antibiotic level



Figure 4.12: Product Requirement Document for overall level

- 1) Provide users with 3 dimensions of data page and menu, respectively are pair Level Page, antibiotic level page and overall page
- 2) Pair Level Page: Allows the user to drop down to select microorganism, antimicrobial and specimen type to view the data visualization content under the certain selections. The contents include number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair and the resistance time-series for a pair.
- 3) antibiotic level page: Shows the distribution of all antibiotics under a species for the user
- 4) Overall Level Page: Show the user the overall distribution of all specimens at different levels from an overall perspective.

4.3 Introduction to the Development Environment.

Based on the demand analysis in Chapter 3, the following technology stack is currently used for technology research and development:

- 1) Python 3.7: It mainly uses the data processing ability of Python to clean and preprocess the data of the source file CSV. And on the premise of completing the first two steps, call Pyamr. Core library to carry on the data analysis of core indicators.
- 2) MongoDB: database development tool, data processing and analysis results are stored in the database for the convenience of server invocation.
- 3) Node server: It provides restful api based on Node.js and Egg.js framework and mainly provides interfaces for querying indicators of different pairs (for example, querying the Sari timeline sequence diagram of a certain microorganism, a certain antimicrobial and a certain specimen).
- 4) Echart: Based on the ECHART data visualization component library, the data is transformed into corresponding charts, such as line chart, bar chart, pie chart, word cloud chart, etc.
- 5) HTML + CSS +javascript VUE: Based on the the web development framework VUE, using modern the website technology HTML5 + CSS3 +javascript ES6, the project can quickly build the website pages to interact with users, and show the above data situation and data visualization charts.

Therefore, the development environment of this project is mainly composed of Python3.7 +MongoDB+ Node.js + Echart + HTML + CSS + Javascript + VUE.

Chapter 5

System implementation

5.1 Overall system implementation scheme

The overall implementation scheme of the system is shown in Fig 5.1 below, which is divided into 3 layers and 6 parts:

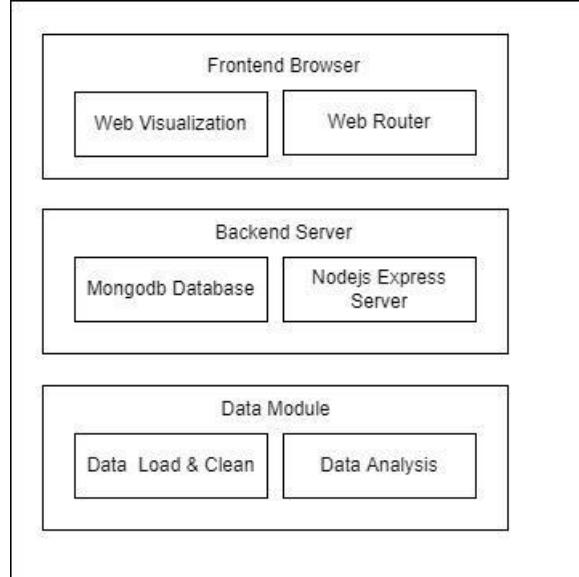


Figure 5.1: Three layers of the overall system implementation

- 1) The third layer is the data module, mainly including data loading, cleaning and data analysis. It is implemented based on Python, this module will output data analysis results. CSV to the upper layer.

- 2) The second layer is the back-end server module, mainly including data storage and data API implementation. The data storage part is implemented by MongoDB, which accepts the CSV produced by the data module and converts it into the database table storage. The API implementation is based on Node.js Express, which will read the corresponding data of MongoDB database according to the API request parameter generated by users and return JSON data to the front-end.
- 3) The first layer is the front-end interaction layer, which is also the browser business layer. It mainly includes user interface and interactive routing, as well as a separate web data visualization module. After the user selects an interaction on the interface, restful API is called to obtain the returned result. Through web visualization Echarts, the result is assembled into corresponding visual components (such as line charts, bar charts, and tree charts) to display to the user.

5.2 Data processing module implementation

The data processing module is mainly composed of data cleaning and meta data generation, which are implemented by Python as follows:

- 1) Import the source file. CSV using pd.read_csv
- 2) Then, the CLWSQL008 method is used to clean the data and find the fields where all fields are normal
- 3) Then, dTYPE is used to map the fields. The mapping relationship is as Fig 5.2,

```

    Returns
    ----
    """
    # Constants
    # -----
    # This dtype is important, specially for the
    # patient_nhs_number that otherwise will give
    # some problems (number poor visualisation).
    dtype = {
        'patient_nhs_number': object # it does not exist but PtNumber does.
    }

    rename = {
        'DiagnosticTestID': 'uuid',
        'PtNumber': 'patient_id',
        'Academic': 'laboratory_number',
        'BatchTestCode': 'specimen_code',
        'OrderName': 'specimen_name',
        'SpecType': 'specimen_description',
        'OrgPieceCounter': 'microorganism_piece_counter',
        'OrgCode': 'microorganism_code',
        'Organism': 'microorganism_name',
        'DrugCode': 'antimicrobial_code',
        'AntiBiotic Name': 'antimicrobial_name',
        'SensMethod': 'sensitivity_method',
        'Sensitivity': 'sensitivity_code',
        'MIC': 'mic',
        'Reported': 'reported',
    }

```

Figure 5.2: The mapping relationship in the file

- 4) Remove duplicate data
- 5) Finally, clean analysis data can be obtained

5.3 Data analysis module implementation

5.3.1 Pair Level

Sari is the degree of resistance of a particular microorganism to a particular antimicrobial in the same sample type. Typically, SARI is equal to the total number of resistant and intermediate divided by the total number of resistant, intermediate and sensitive.

In order to help doctors better analyze AMR data, this project plotted the line chart of Sari value over time under ITI and OTI respectively. In the calculation methods of ITI and OTI, it is necessary to define the test frequency and the period to which the samples used in the calculation of SARI belong. Where, under the ITI method, frequencies are equal to time periods, that is, they do not overlap. In the OTI method, the set frequency should be less than the time period. With different frequencies and time periods set, different SARI timings can be plotted and compared.

5.3.2 Antibiotic Level

ASAI shows different antimicrobials are effective on which range of microorganisms. ASAII was calculated under the dimension of genus, using SARI the projectighted values of species dimension. To calculate ASAII, firstly select samples of the same type. Secondly, the calculation method of the ASAII value of a certain antibiotic is the the projectighted average of the sari of all microorganisms in all genus dimensions under a certain antibiotic. Among them, the calculation method of the sari value of microorganisms in a genus dimension, is the the projectighted average (the projectighted by the number of samples) of the effective sari of microorganisms in all species dimensions in this genus. When the SARI value is less than the specified value, it is considered that the antibiotic has a therapeutic effect on the bacteria, that is, the effective SARI value mentioned above.

5.3.3 Overall Level

Overall Level Displays the overall Treemap image. The doctor can get the following information by hovering over the mouse, antimicrobial code, the number of samplesmicroorganism code specimen code microorganism name, antimicrobial name, asri hard, sari medium and sari soft.

5.4 Data visualization module implementation

From 5.3, the project can get data indicators of different levels. Through assembly and stitching, the project can analyze their time ductility and cohesion.

Specifically, according to the transformation of different indicators in a period of time, the trend can be analyzed through time series chart and line chart. The Gram-positive and Gram-negative of different specimens in various dimensions can be analyzed through aggregation analysis and horizontal bar chart. The overall distribution can be obtained through Treemap word cloud chart.

Based on this, the project's data visualization module mainly focuses on line chart, bar chart, pie chart, word cloud chart. In terms of technology selection, this project realizes data visualization of two languages, namely data visualization based on Python plot library and data visualization based on Echarts js library.

5.4.1 Data visualization based on Python Plot

Taking Time Series Line Graph as an example, the realization of data frame data visualization based on Python plot is divided into the following steps:

- 1) Install and reference the plot and SNS libraries in Python files
- 2) Set the seaborn style using the sns.setStyle method
- 3) Create an instance of Figure and axie with the plot.subplots method
- 4) Set graph-related parameters, using sns.line plot to set the line chart style (width, title, bar scale under the line chart, transparency, etc.)
- 5) Import the data into the Figure instance

6) The `plot.tight_layout` method is called to implement the drawing. The specific code is as Fig 5.2.

```

27 # Main
28 # -----
29 # Load data
30 data, antibiotics, organisms = load_data_nhs()
31
32 # Count records per specimen code
33 specimen_code_count = data \
34     .groupby('laboratory_number').head(1) \
35     .specimen_code.value_counts(normalize=True) \
36     .sort_values(ascending=False)
37
38 # Filter most frequent specimens
39 data = data[data.specimen_code.isin( \
40     specimen_code_count.index.values[:1])]
41
42 # -----
43 # Compute Freq and SARI
44 # -----
45 # Create instance
46 freq = Frequency(column_antibiotic='antimicrobial_code',
47                   column_organism='microorganism_code',
48                   column_date='date_received',
49                   column_outcome='sensitivity')
50
51 # Create daily frequency
52 daily = freq.compute(data, strategy='ITI',
53                       by_category='pairs',
54                       fs='1M')
55
56 daily['freq'] = daily.sum(axis=1)
57
58 import plotly.express as px
59 import pandas as pd
60
61 print(daily)
62
63 daily = daily.reset_index()
64 daily = daily[daily.SPECIE.isin(['ECOL', 'PSEUD']) &
65               | daily.ANTIBIOTIC.isin(['AAUG'])]
66
67
68
69 fig = px.bar(daily, x='DATE', y='freq',
70               title='Time Series with Range Slider and Selectors',
71               facet_col='SPECIE')
72
73 fig.update_xaxes(
74     rangeslider_visible=True,
75     rangeslider=dict(
76         buttons=list([
77             dict(count=1, label="1m", step="month", stepmode="backward"),
78             dict(count=6, label="6m", step="month", stepmode="backward"),
79             dict(count=1, label="YTD", step="year", stepmode="todate"),
80             dict(count=1, label="1y", step="year", stepmode="backward"),
81             dict(step="all")
82         ]))
83 )
84
85 )
86
87 fig.show()
88 fig.write_html("../../../docs/source/_static/htmls/{}.html" \
89 | .format('dai-plot_pyplot_timestries'))
90

```

Figure 5.3: The visualization code in python plot

The effect is as Fig 5.4.

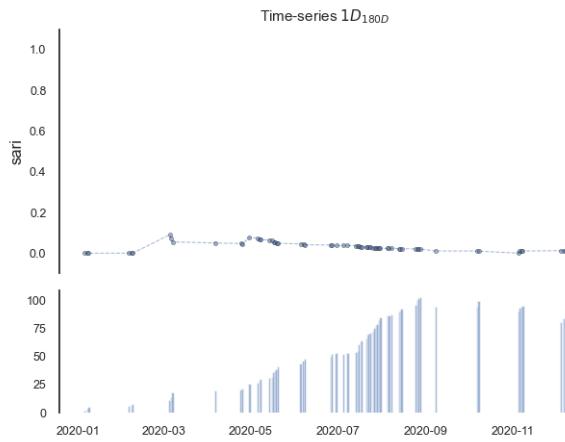


Figure 5.4: The visualization of SARI time-series in a pair

5.4.2 Data visualization based on Echarts Visualization

Echarts is an open source JavaScript visualization library that provides rich open source components for data-driven visualization, as shown in the following Fig 5.5.

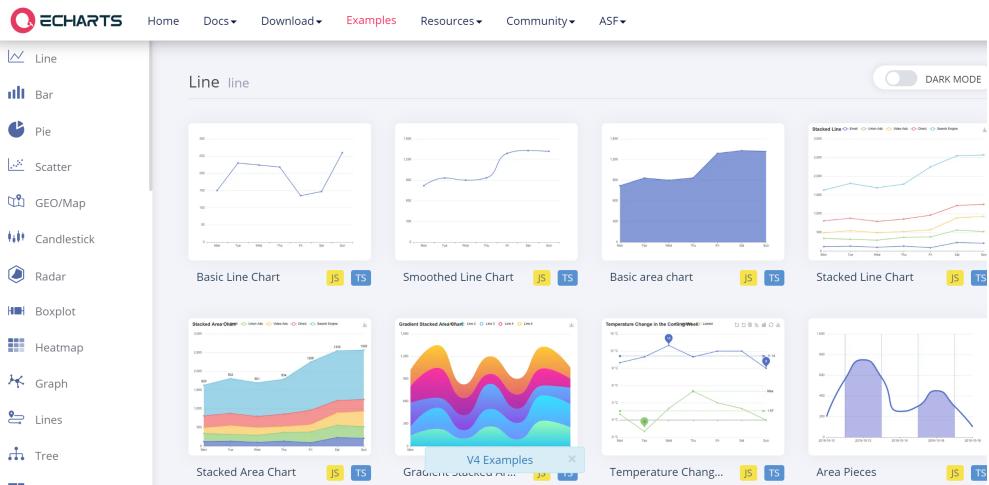


Figure 5.5: Examples of graphs in Echarts

Next, the bar graph is taken as an example to introduce the realization of data visualization based on ECHART, which is divided into the following steps:

- 1) To install and introduce the EcHART library, the project can install Echarts via NPM
install – Save Echarts, and through
import * as echarts from 'echarts'

Of course, the compiled EChart.js can be directly referenced by script in HTML Head through the CDN.

2) Define an instance of EcHart in javascript code and register it to a DOM node with the init method, such as

```
echarts.init(document.getElementById('main'));
```

3) Set the graphical visualization option of the instance, and set the title/tooltip/type/data by configuring JSON to render the specific instance on the the web side.

The implementation code is as Fig 5.6:

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>ECharts</title>
    <!-- 引入刚刚下载的 Echarts 文件 -->
    <script src="echarts.js"></script>
  </head>
  <body>
    <!-- 为 Echarts 准备一个定义了宽高的 DOM -->
    <div id="main" style="width: 600px;height:400px;"></div>
    <script type="text/javascript">
      // 由于准备好的dom，初始化echarts实例
      var myChart = echarts.init(document.getElementById('main'));

      // 指定图表的配置项和数据
      var option = {
        title: {
          text: 'ECharts 入门示例'
        },
        tooltip: {},
        legend: {
          data: ['销量']
        },
        xAxis: {
          data: ['衬衫', '羊毛衫', '雪纺衫', '裤子', '高跟鞋', '袜子']
        },
        yAxis: {},
        series: [
          {
            name: '销量',
            type: 'bar',
            data: [5, 20, 36, 10, 10, 20]
          }
        ]
      };

      // 使用刚指定的配置项和数据显示图表。
      myChart.setOption(option);
    </script>
  </body>
</html>
```

Figure 5.6: The specific visualization code in Echarts

The effect is as Fig 5.7:



Figure 5.7: The visualization of Echarts example

5.5 The website Implementation

5.5.1 Database implementation and table design

The database uses MongoDB non-relational database to realize the related database logic. The database name is AMR_analysis and contains 5 tables. The table structure is shown in Fig 5.8:

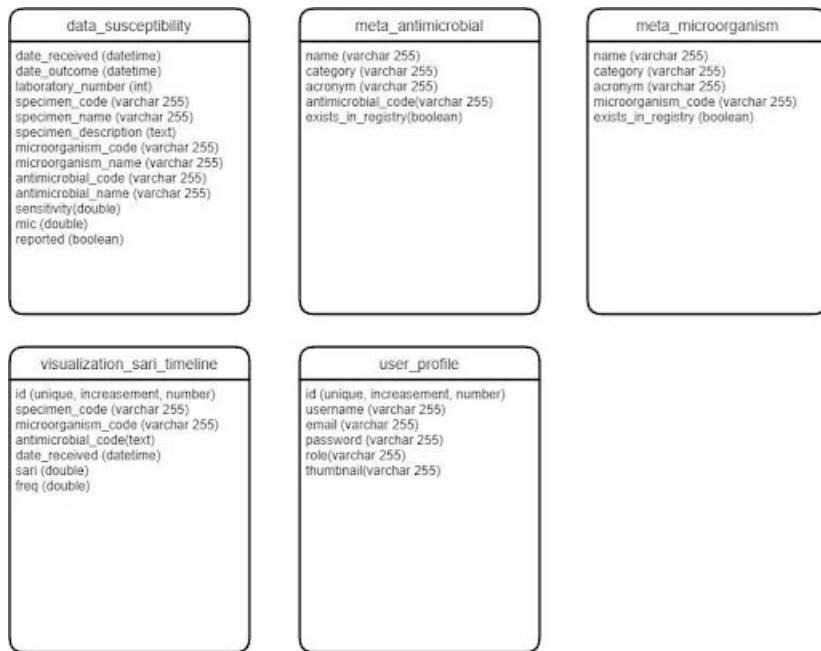


Figure 5.8: The table structure in MongoDB

The data table contains the following 5 tables, and the design and implementation are introduced separately in the following section.

1. Data_susceptibility: the original data table that stores all effective data after data cleaning. The susceptibility field information is as follows

Date_received: indicates the data acquisition time, which is used as the X-axis analysis in subsequent time series analysis

Date_outcome: data production time, used as backup data only

Laboratory_number: laboratory number, used as backup data only

Specimen_code: sample code, unique identifier for representative samples, the follow-up in multiple dimensions will include specimen analysis

Specimen_name: the name of the sample meta information, as a specimen of the display information

Specimen_description: sample description meta information, as a specimen of the display information

Microorganism_code: microorganism code, which represents this only identifier, and actually includes this in multiple analysis dimensions

Microorganism_name: microorganism name of this meta information, and this is actually just a showcase message

Antimicrobial_code: antimicrobial code, used to represent the unique identifier of antimicrobial, the follow-up in multiple dimensions will include antimicrobial analysis

Antimicrobial_name: the name of the meta information antimicrobial, as antimicrobial display information

Sensitivity: the sensitivity of the record (i.e., the pair)

Mic: The MIC of the line record (that is, a pair)

Reported: whether bank records have been reported

After importing the data, the table has 47,133 rows of data.

2. Meta_antimicrobial: an antimicrobial data information table contains the following fields:

Name: name of antimicrobial

Category: antimicrobial meta information, indicating the domain

Acronym: abbreviation

Exists_in_registry: Whether it is registered

Antimicrobial_code: the unique identifier of the antimicrobial

After importing the data, the table has 61 rows of data

3. Meta_microorganism: Data information sheet, will just be the following field information:

Name: name of microorganism

Category: this is actually just microorganism meta information, indicating the user field

Acronym: abbreviation

Exists_in_registry: whether it is registered

Microorganism_code: the unique identifier of the microorganism

After importing the data, the table has 61 rows of data

4. Visualization_sari_timeline: line chart data information table, the field information is as follows:

Id: the id

Specimen_code: sample code, unique identifier for representative sample

Microorganism_code: actually this is just the code, which represents just a unique identifier

Antimicrobial_code: antimicrobial code, used to represent antimicrobial unique identifier

Date_received: indicates the data acquisition time, which is used as the X-axis analysis in subsequent time series analysis

Freq: how often this line records (i.e. a pair)

Sari: this row records the SARI value of a pair

5. User_profile: user information table for simple processing of user logins and user information

Id: the id

Username: indicates the username

Email: indicates the user email address

Password: indicates the user password

Role: indicates the user role. The default value is admin

5.5.2 Back-end service implementation

Implementation of the Back-end HTTP Server

The back-end Httpserver is mainly based on the NodeJS Express library to build the HTTP server.

Express is an NPM Node package designed to quickly build Node.js the web Servers and provide apis. The implementation method is as follows:

1. First, define the the webServer folder and run NPM init to initialize the Node.js project package, as shown in the Fig 5.9.

```
→ webserver npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (webserver)
version: (1.0.0)
description: webserver for py-amr
entry point: (index.js)
test command:
git repository:
keywords: py-amr
author: jinyujia
license: (ISC)
About to write to /Users/david-dai/test/jinyujia/webserver/package.json:

{
  "name": "webserver",
  "version": "1.0.0",
  "description": "webserver for py-amr",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [
    "py-amr"
  ],
  "author": "jinyujia",
  "license": "ISC"
}

Is this OK? (yes)
```

Figure 5.9: Initialize the NodeJS project package

2. Install the Express package and save the dependencies by using NPM install – Save Express
3. Create server.js, reference the Express package, and specify the HttpServer listening port and API route as Fig. 5.10,

```
var express = require('express');
var app = express();

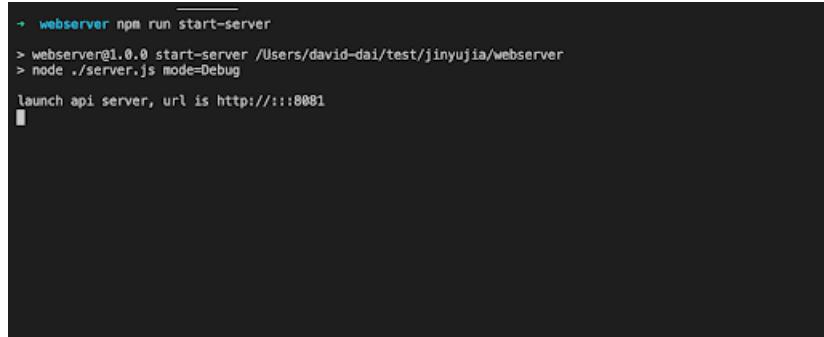
app.get('/', function (req, res) {
  res.send('Hello World');
})

var server = app.listen(8081, function () {
  var host = server.address().address
  var port = server.address().port

  console.log(`[\"launch api server, url is http://${host}:${port}\"]`)
})
```

Figure 5.10: Specify the HttpServer listening port and API route

4. Modify the package.json script in the root directory and add the start command
"start-server": "node./server.js mode=Debug"
5. Enter the APP, install all the dependency packages of API-Server, and run NPM run start-server to start the service. As shown in Fig 5.11, the service is started.



```
→ webserver npm run start-server
> webserver@1.0.0 start-server /Users/david-dai/test/jinyujia/webserver
> node ./server.js mode=Debug
launch api server, url is http://:::8081
```

Figure 5.11: Run NPM run start-server to start the service

In this case, the project can use the configured port 3000 and test API route to test whether the service is normal as Fig 5.2.



Figure 5.12: ‘Hello World’ Test result

Connecting to the MongoDB Database

The back-end service needs to query the MongoDB database, which is mainly achieved through the NPM MongoDB package. The implementation method is as follows: 1. Install and reference the MongoDB package by NPM install -save MongoDB and by require('MongoDB')

2. Define MongoDB config URL and create connection using MongoClient.connect
3. After the connection is successful, use the dbo.collection("site").find (where-Str).toArray () method to query the result and convert it into an array as Fig 5.13.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("runoob");
  var whereStr = {"name":'test'}; // query
  dbo.collection("site").find(whereStr).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

Figure 5.13: Query the result in MongoDB

Restful api implementation

This article focuses on the Timeline query interface, which consists of two interfaces: QueryTotalCount the project can query the total number of indicators in a pair Query-TimelineData is used to query the SARI value of a pair within a specified period of time.

The following uses /queryTimelineData as an example to illustrate the API implementation.

1. Define the request body parameters, according to the requirements of user demand from microorganism/antimicrobial/specimen drop-down box to choose to call interface, therefore the request body parameters are defined as follows:

```
{
  specimen_code: '',
  microorganism_code: '',
  antimicrobial_code: ''}
```

2. Define the route, express. Post ('/queryTimelineData', function(req,res)) to define the

route, and define the apiHandler method of the route

3. Define process routing apiHandler, get into the pair of parameter values, the inside is specimen_code respectively, microorganism_code, antimicrobial_code
4. Connect to MongoDB, search for the rows that match the conditions in the data_susceptibility table, and return the result as Fig 5.14.

```
app.post('/queryTimelineData', function(req, res) {
  //fetch request params
  const {specimen_code, microorganism_code, antimicrobial_code} = req

  // query db
  queryDb(specimen_code, microorganism_code, antimicrobial_code).then((data)=>{
    // set call back data to response data
    res.json({
      data
    })
  })
})
```

Figure 5.14: Search for the rows that match the conditions in the data_susceptibility table

5. Use Postman to test whether the interface returns normally as Fig 5.15.

The screenshot shows the Postman interface with the following details:

- Network Tab:** Selected tab.
- Request URL:** `/queryTimelineData`
- Headers:** None listed.
- Preview Tab:** Selected tab, showing the JSON response structure.
- Response Body (JSON):**

```
{
  "data": [
    {
      "field1": "0",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN",
      "date_received": "2020-01-07",
      "freq": "1.0",
      "microorganism_code": "SMIL",
      "sari": "0.0",
      "sensitive": "1.0",
      "specimen_code": "WOUCL"
    },
    {
      "field1": "1",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    },
    {
      "field1": "2",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    },
    {
      "field1": "3",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    },
    {
      "field1": "4",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    },
    {
      "field1": "5",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    },
    {
      "field1": "6",
      "specimen_code": "WOUCL",
      "microorganism_code": "SMIL",
      "antimicrobial_code": "AVAN"
    }
  ]
}
```

Figure 5.15: Test result

5.5.3 Front-end implementation

Front-End MVVM Framework Integration

MVVM is short for Model-View-ViewModel. Model stands for data Model, where the project can also define the business logic for data modification and manipulation. The View represents the UI component, which is responsible for translating the data model into the UI and presenting it. The ViewModel listens for changes in Model data and controls View behavior, handles user interaction, and is simply understood as an object that synchronizes View and Model, connecting Model and View. Under the MVVM architecture, the View and Model are not directly related, but interact through the ViewModel. The interaction between the Model and ViewModel is two-way, so the changes of View data will be synchronized to the Model. Changes in Model data are immediately reflected in the View. The ViewModel connects the View layer to the Model layer through bidirectional data binding, and the synchronization between the View and Model is completely automatic and hands-free, so developers only need to focus on the business logic, do not need to manually manipulate the DOM, do not need to focus on data state synchronization issues, Complex data state maintenance is completely managed by MVVM.

Front-end Template AdminLTE

Adminlte is an open source front-end component library, can be based on his fast build data management platform, he provides commonly used components such as the website login component, left menu component, LOGO component, form component, drop-down component, table component, and support the extension of visual component library EChart.js. This project mainly uses the login component, menu component, drop-down selection component, and extended support for ECHART data visualization component.

Implementation and effect diagram of each front-end module

1. Pair level module

The user of Pair Level is a data administrator. It selects different pairs based on user requirements and displays the visual data in a pair. The functions of pair Level include: Drop-down choice specimen, the drop-down choose antimicrobial, drop-down selection microorganisms, shows number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair and the resistance time-series for a pair. Its implementation effect diagram is as Fig 5.16:

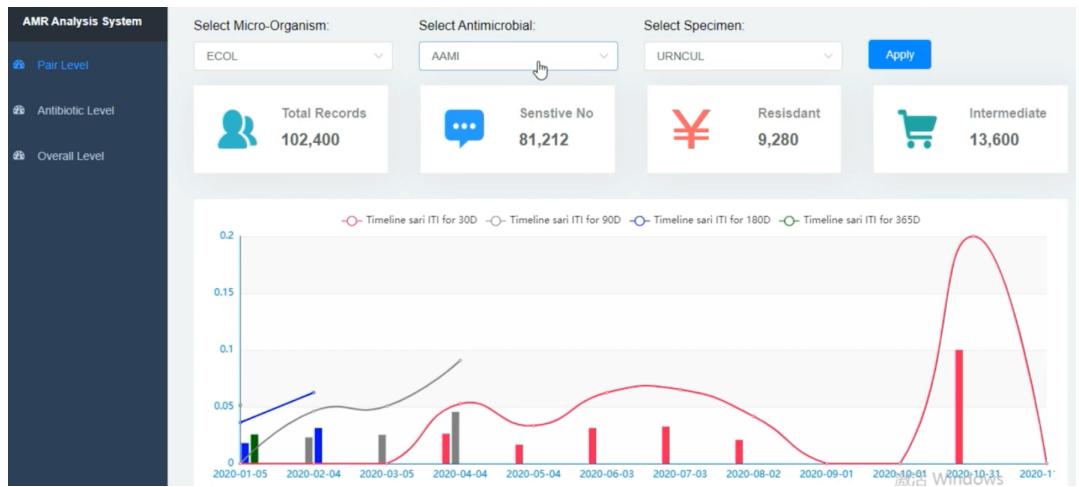


Figure 5.16: Webpage for pair level

2. Antibiotic level module

The user of antibiotic level is the data administrator, its function is mainly according to the needs of the user, choosing different specimens. Thus display the spectrum of activity of antimicrobials bar graph of different specimens under Gram-negative and Gram-positive bacteria. Its implementation effect diagram is as Fig 5.17:



Figure 5.17: Webpage for antibiotic level

3. Overall level module

The user of Overall Level is the data manager, whose role is mainly from the overall perspective to view the overall situation of the three dimensions, as well as the distribution of its layer by layer dimension. The visualization part uses the way of Treemap, and can view the distribution of different dimensions through Hover. Its implementation effect diagram is as Fig 5.218:

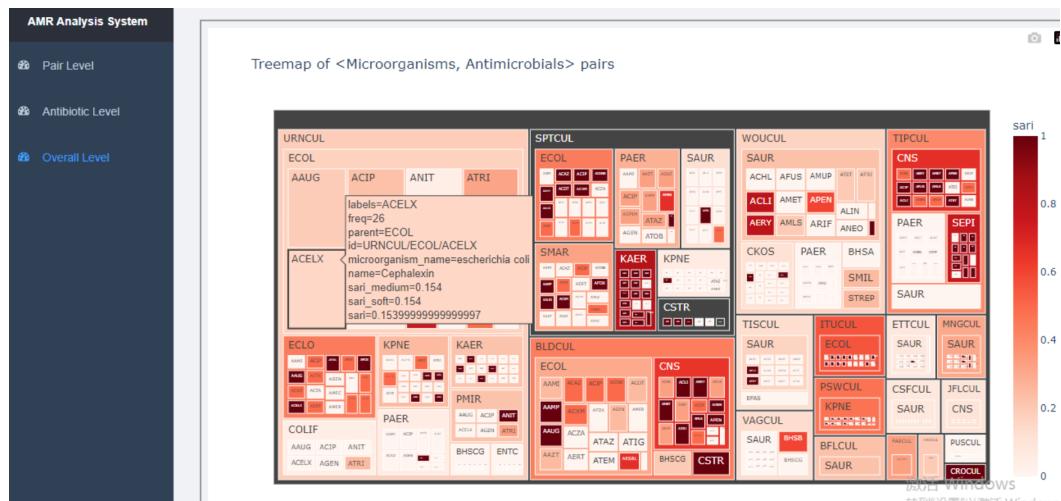


Figure 5.18: Webpage for overall level

4. Other User Modules

In addition to the data visualization module, the article

also implements a simple user login and user information page. The user information page uses el-form to submit the login form in the front end. After verifying the user's account and password is correct, the user can enter the background system. On the user information page, the project can use cookies to authenticate the user and call getInfo to obtain and display the profile information of the user. Its implementation effect diagram is as Fig 5.19 and Fig 5.20:

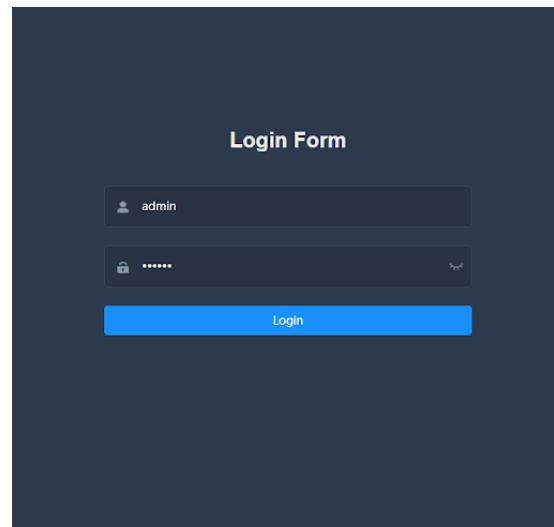


Figure 5.19: Login page

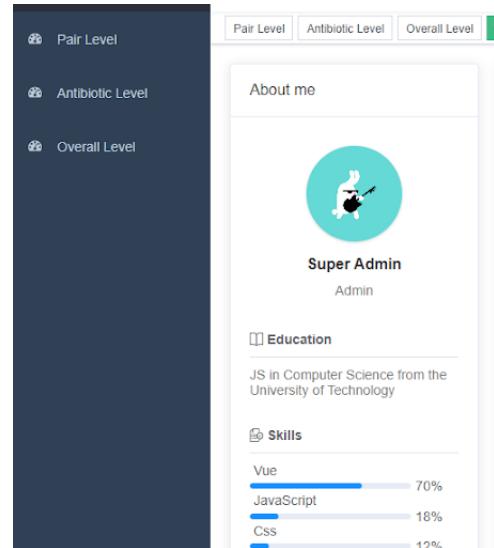


Figure 5.20: User profile page

Chapter 6

System test

6.1 Logging in to a module

6.1.1 Test Purpose

The purpose of this experiment is to test whether the functions of the login interface of the platform meet expectations under different circumstances for customers with different roles.

6.1.2 Experimental Environment and Configuration

Network environment: Windows 10 Software and hardware environment: Chrome 27.0.1453.110m

6.1.3 Test Procedure and Result

1. [The user name and password are incorrect] test Test input: Enter an incorrect user name Expect the output: Please enter the correct user name The actual output: Please enter the correct user name, as shown in Fig 6.1.

Test results: As expected

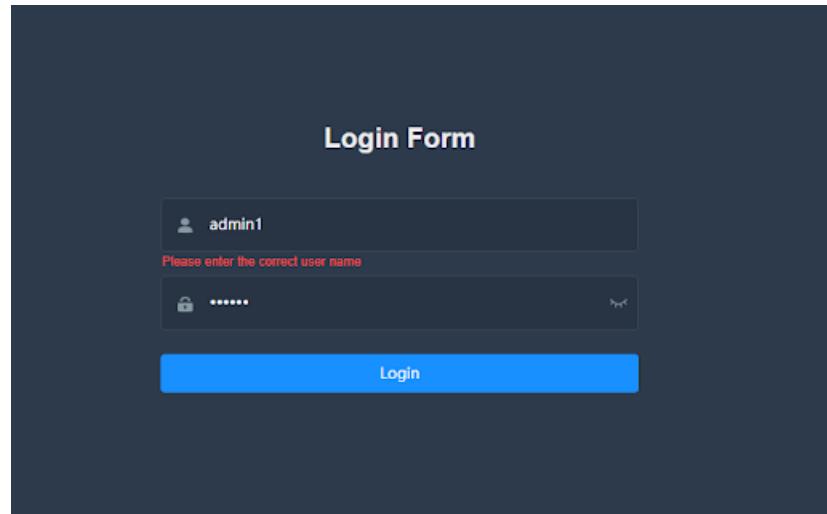


Figure 6.1: Test for incorrect user name

2. [Username and password mismatch] test Test input: Enter an incorrect user name or password Expected output: Password not matched Actual output: Password not matched, as shown in Fig 6.2.

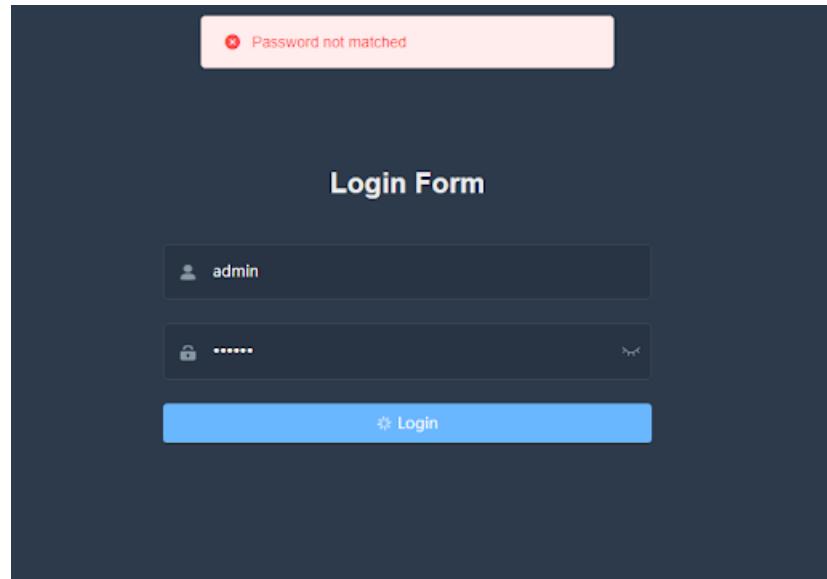


Figure 6.2: Test for incorrect user name

Test results: As expected

3. [The user name and password match successfully] Test Test input: Enter the correct user name or password The message Successfully login is displayed Actual output:

Successfully login is displayed, as shown in Fig 6.3.

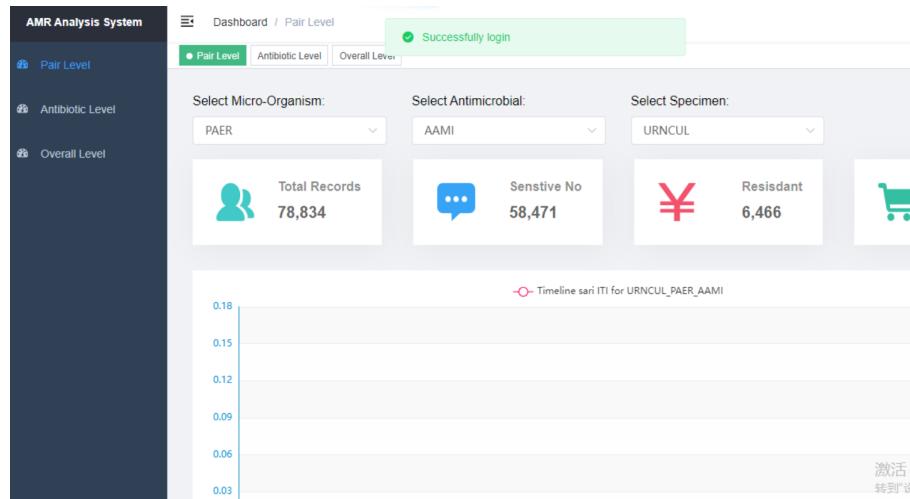


Figure 6.3: Test for correct name and password and successfully login

Test results: As expected

6.2 Pair Level function test

6.2.1 Test Purpose

The purpose of this experiment is to test whether the platform can obtain the corresponding total figure and line figure after the pull-down selection operation for the user's operation behavior at Pair Level.

6.2.2 Experimental Environment and Configuration

Network environment: Windows 10 Software and hardware environment: Chrome 27.0.1453.110m

6.2.3 Test results and analysis

1. [Dropdown selection] test Test input: user mouse focus in 3 drop-down boxes Expect output: the project see a list of options with three drop-down selection boxes

Actual output: the project see a list of options with three drop-down selection boxes, as Fig 6.4.

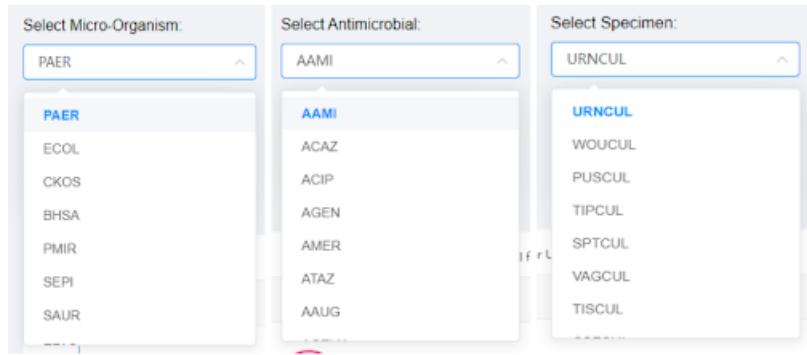


Figure 6.4: Three drop-down lists

2. [Visual chart] Test After good test inputs: the user to select a set pair (specimen/antimicrobial/organism) Expected output: Total number of items/line chart Actual output: Total number of items/line chart, as Fig 6.5.

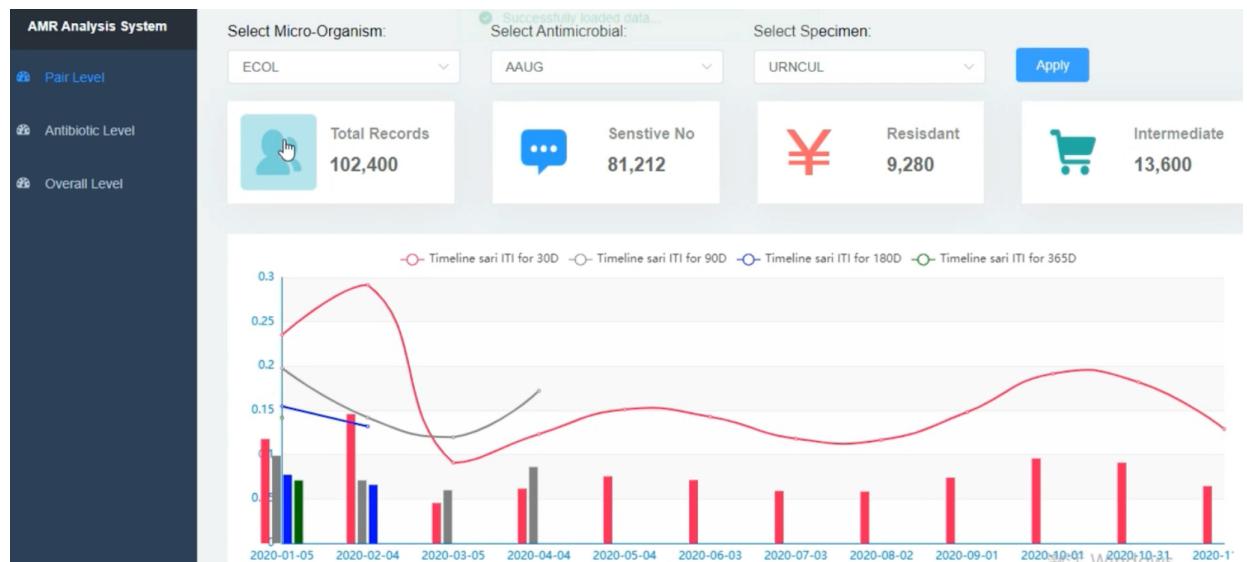


Figure 6.5: Test for pair level

6.3 Antibiotic Level function test

6.3.1 Test Purpose

The purpose of this experiment is to test if the platform can obtain the corresponding horizontal bar chart for the user's operation behavior in Timlel, after using dropdown selection.

6.3.2 Experimental Environment and Configuration

Network environment: Windows 10 Software and hardware environment: Chrome 27.0.1453.110m

6.3.3 Test results and analysis

1. [Visual chart] Test Test input: user selects good specimen Expected output: Horizontal bar chart Actual output: Horizontal bar graph, as Fig 6.6.

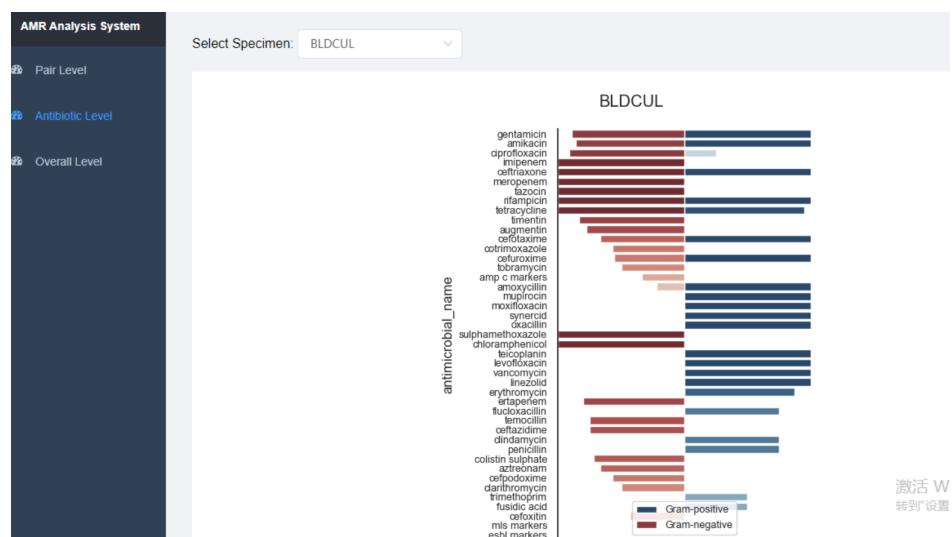


Figure 6.6: Test for antibiotic level

6.4 Overall Level functional test

6.4.1 Test Purpose

The purpose of this experiment is to test whether the platform can obtain the corresponding Treemap and its floating box after the hover operation, according to the user's operation behavior at Overall Level.

6.4.2 Experimental Environment and Configuration

Network environment: Windows 10 Software and hardware environment: Chrome 27.0.1453.110m

6.4.3 Test results and analysis

1. Visual chart Test Test input: None Expected output: The overall Treemap diagram Actual output: The overall Treemap diagram, as Fig 6.7.

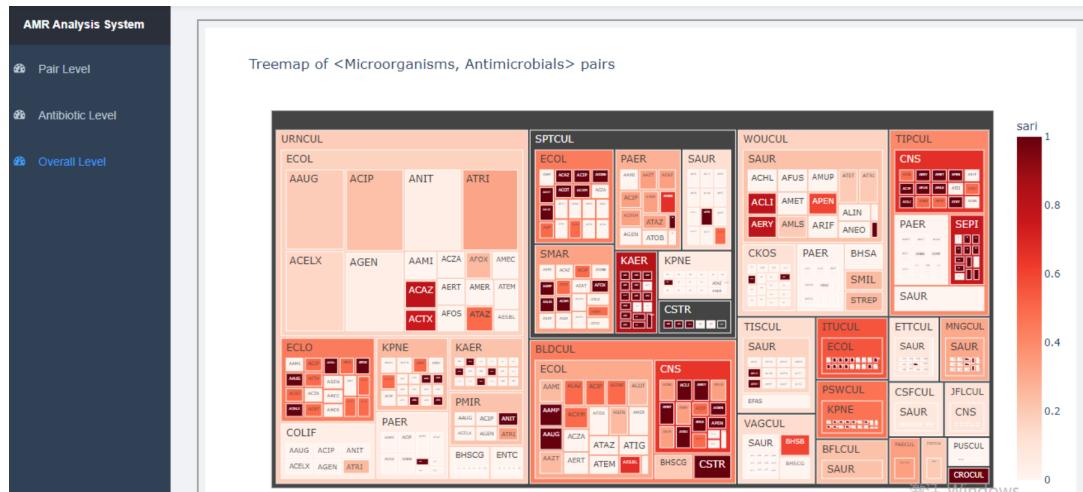


Figure 6.7: Test for overall level

2. [Hover display floating box] test Test input: The user hover on Treemap Expected output: Float box showing details of different pairs Actual output: Float boxes showing details of different pairs are displayed, as Fig 6.8.

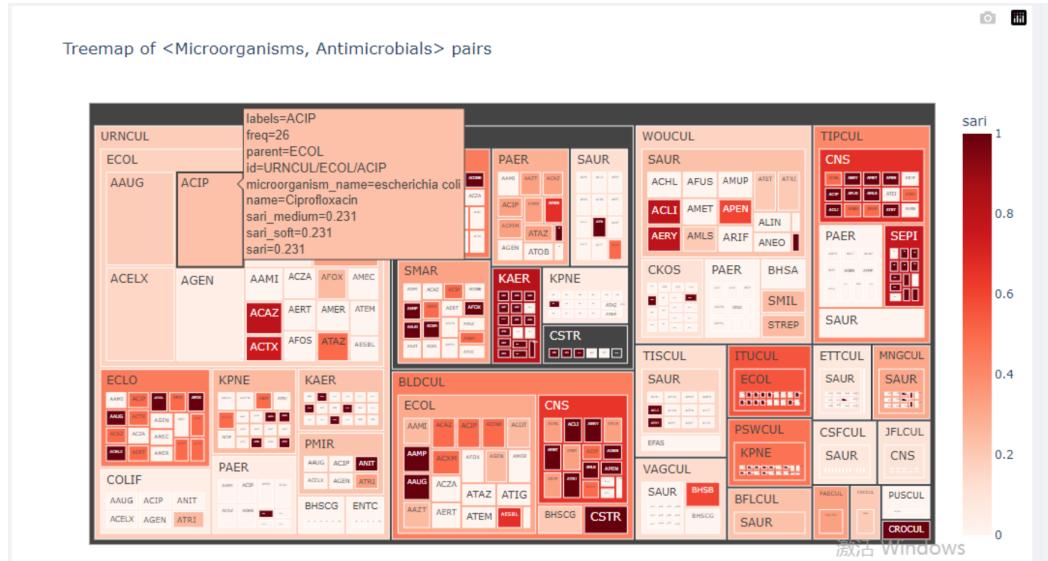


Figure 6.8: Test for hover display floating box

Chapter 7

Discussion and conclusions

7.1 Discussion

The problem of AMR is one of the difficult problems faced by human medicine today, and various countries and organizations are seeking solutions. One of the most important causes of the AMR problem is the inappropriate and excessive use of antimicrobials by clinicians and pharmacists. This project builds an AMR monitoring web page system that can interact with users to provide doctors with required data, charts and reference suggestions.

It is of great significance to collect sensitivity test data to help quantify the AMR indicators and give doctors accurate information of the AMR situation in the area. This project uses SARI and ASAI to carry out the processing and analysis of the data, providing three level information, pair level, cleaver level and overall level. Pair Level contains the number of total records per interval for a pair, number of sensitive/resistant/intermediate records for a pair and multiple ITI and OTI resistance time-series for a pair. Antibiotic Level contains a lateral bar chart of ASAI. Overall Level contains the overall treemap. Among them, the time series of SARI is visualized with ITI and OTI, which solves the problem of granularity and accuracy. Especially when the data is insufficient, more practical AMR information can still be obtained through OTI. When designing and building the interactive AMR monitoring web page system, this project

designed a three-layer and six-part structure. First, process the original data and store all the data in MongoDB. Then, Nodejs is used to build Node Server and provide restful api. Finally, the analysis results are displayed on the website using the Web visualization gallery Echart.

This project also has some limitations.

1. Support for real-time external databases. This project only uses data of susceptibility from 2020-01 to 2020-12. Later from the perspective of practicality, the project can expand the support of custom data in a wider range of time dimensions, and also consider accessing a real-time external database to realize real-time data analysis and data visualization of susceptibility data.
2. Add more dimensions of data analysis and data visualization to explore more data value. From the perspective of data analysis, data types and indicators can be added, such as patients' underlying diseases, feedback on subsequent treatment effects, etc. The data visualization perspective can add more visual legends, such as pie charts, heat maps, and other graphs.

7.2 Conclusions

The processing, analysis and visualization of sensitivity test records is an effective solution used by countries to mitigate the AMR problem. Based on the collection, processing and visualization of sensitivity test records, this project provides ITI and OTI multiple resistance time-series for a pair, a lateral bar chart of ASAII and the overall treemap. This paper mainly designs and implements a three-layer and six-part AMR monitoring web page system which can interact with users. The system uses Python to process and analyze data, MongoDB to store data, Node.js to build Node Server and Echart visualization gallery to show the above AMR information on the website. This system can provide doctors with clear and accurate AMR key information and clinical recommendations, which can alleviate the rapidly developing AMR problem to a certain extent.

Bibliography

- [1] J. Conly and B. Johnston, “Where are all the new antibiotics? the new antibiotic paradox,” *Canadian Journal of Infectious Diseases and Medical Microbiology*, vol. 16, no. 3, pp. 159–160, 2005.
- [2] R. Safdari, M. GhaziSaeedi, H. Masoumi-Asl, P. Rezaei-Hachesu, K. Mirnia, and T. Samad-Soltani, “A national framework for an antimicrobial resistance surveillance system within iranian healthcare facilities: towards a global surveillance system,” *Journal of global antimicrobial resistance*, vol. 10, pp. 59–69, 2017.
- [3] S. J. Hoffman, K. Outterson, J.-A. Røttingen, O. Cars, C. Clift, Z. Rizvi, F. Rotberg, G. Tomson, and A. Zorzet, “An international legal framework to address antimicrobial resistance,” pp. 66–66, 2015.
- [4] R. Lawrence and E. Jeyakumar, “Antimicrobial resistance: a cause for global concern,” in *BMC proceedings*, vol. 7, no. 3. BioMed Central, 2013, pp. 1–14.
- [5] C. F. Amábile-Cuevas, *Antimicrobial resistance in bacteria*. Horizon Scientific Press, 2007.
- [6] R. S. Evans, J. A. Olson, E. Stenehjem, W. Buckel, E. Thorell, S. Howe, X. Wu, P. Jones, and J. Lloyd, “Use of computer decision support in an antimicrobial stewardship program (asp),” *Applied clinical informatics*, vol. 6, no. 01, pp. 120–135, 2015.
- [7] S. Wallace and N. Damani, “O022: Development of an electronic dashboard to assist surveillance,” *Antimicrobial Resistance and Infection Control*, vol. 2, no. 1, pp. 1–1, 2013.

- [8] R. D. Stedtfeld, M. R. Williams, U. Fakher, T. A. Johnson, T. M. Stedtfeld, F. Wang, W. T. Khalife, M. Hughes, B. E. Etchebarne, J. M. Tiedje *et al.*, “Antimicrobial resistance dashboard application for mapping environmental occurrence and resistant pathogens,” *FEMS microbiology ecology*, vol. 92, no. 3, 2016.
- [9] M. K. Obenshain, “Application of data mining techniques to healthcare data,” *Infection Control & Hospital Epidemiology*, vol. 25, no. 8, pp. 690–695, 2004.
- [10] H. Joudaki, A. Rashidian, B. Minaei-Bidgoli, M. Mahmoodi, B. Geraili, M. Nasiri, and M. Arab, “Using data mining to detect health care fraud and abuse: a review of literature,” *Global journal of health science*, vol. 7, no. 1, p. 194, 2015.
- [11] B. K. Arya, D. Robert, S. D. Bhattacharya, and J. Mukhopadhyay, “A framework for web based geographical information systems for country wide antimicrobial resistance monitoring,” *Health Policy and Technology*, vol. 2, no. 2, pp. 85–93, 2013.
- [12] S. Galvin, N. Bergin, R. Hennessy, B. Hanahoe, A. W. Murphy, M. Cormican, and A. Vellinga, “Exploratory spatial mapping of the occurrence of antimicrobial resistance in e. coli in the community,” *Antibiotics*, vol. 2, no. 3, pp. 328–338, 2013.
- [13] T. C. Ekundayo and A. I. Okoh, “Antimicrobial resistance in freshwater plesiomonas shigelloides isolates: Implications for environmental pollution and risk assessment,” *Environmental Pollution*, vol. 257, p. 113493, 2020.
- [14] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, “Analyzing data granularity levels for insider threat detection using machine learning,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.
- [15] Ph.D. dissertation.
- [16] J. S. Gerber, A. L. Hersh, M. P. Kronman, J. G. Newland, R. K. Ross, and T. A. Metjian, “Development and application of an antibiotic spectrum index for benchmarking antibiotic selection patterns across hospitals,” *infection control & hospital epidemiology*, vol. 38, no. 8, pp. 993–997, 2017.

- [17] T. Gregersen, “Rapid method for distinction of gram-negative from gram-positive bacteria,” *European journal of applied microbiology and biotechnology*, vol. 5, no. 2, pp. 123–127, 1978.
- [18] L. Manovich, “What is visualization,” *paj: The Journal of the Initiative for Digital. – 2010*, 2010.