

Design and Implementation of Public Opinion Monitoring System Based on Cloud Platform

Wei Wei^{1,2}, Long Wang¹, Xiaowan Li¹, Beibei Zhang¹, Rafał Scherer³

¹ School of Computer Science and Engineering, Xi'an University of Technology, China

² Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an University of Technology, China

³ Czestochowa University of Technology AI, Poland

weiwei@xaut.edu.cn, 862721961@qq.com, endeavour69@163.com, bbzhang115@hotmail.com, rafal.scherer@pcz.pl

Abstract

With the rapid development of the Internet, the network media has been recognized as the "fourth media" after newspapers, radio and television. The network has become one of the main carriers to reflect social public opinion. The public opinion effectiveness of weibo, Zhihu and other new media platforms is also growing, and the detection of public opinion is increasingly difficult to control in time. It is more and more difficult to understand multi platform information in time and synchronously.

Based on Python crawler technology and cloud platform technology to develop the public opinion monitoring system under the cloud platform, so as to achieve timely and accurate public opinion information of multiple media platforms, and give the corresponding public opinion change information in a period of time, which will be more conducive to the control and monitoring of public opinion information by users.

Keywords: Crawler technology, Cloud platform, Multithreading, Data analysis

1 Research and Development Status at Home and Abroad

Nowadays, social groups and organizations increasingly recognize the importance of online public opinion, so a large number of Internet companies and traditional media companies have begun to develop this technology. There are a large number of mature public opinion monitoring systems on the market, such as, Mainly in China: yqt365.com based on Sina Weibo, initially mainly capture all Weibo data, but now also began to support the capture of WeChat, website, video, client and other network source data; yuqing.baidu.com based on Baidu, according to the frequency of keywords used by netizens to search engines to determine the direction of public opinion, and also

supports the capture of data on social platforms, news blogs and e-commerce platforms; there is also the people's public opinion, launched by People's Online under the People's Daily; It is mainly for the public, government platforms and other social and livelihood and government affairs information, and is more credible. Correspondingly, there are a large number of mature public opinion monitoring systems abroad, such as Meltwater (Rongwen), etc.; and provide mature public opinion analysis and forecast supervision.

However, although most mature public opinion monitoring systems can provide mature public opinion monitoring and early warning, but the relatively short-term free experience time and high user fees, so that the users are still limited to the scope of government organizations and large enterprises, it is not a good experience for micro enterprises and individual users, therefore, there is still a big gap for the needs of individual users and micro and small enterprises [1-19].

2 System Planning and Analysis

In the system planning and analysis stage, the main purpose is to understand the specific functions, performance, reliability and other requirements of the software system expressed by the user through investigation and analysis, user communication, etc.; systematize it and transform it into a certain standard to form a complete and clear description. With the normative documents, determine what functions the software needs to achieve and what tasks to complete. In addition, some non-functional requirements of software (such as software performance, reliability, scalability, etc.), constraints of software design, and the relationship between other software at runtime are also the goals of software requirements analysis [20-29].

2.1 System Overview

The system overview mainly introduces the design goals of the system, and roughly introduces some

*Corresponding Author: Wei Wei; E-mail: weiwei@xaut.edu.cn

standards that the public opinion monitoring system under the cloud platform needs to achieve; the system outline composition mainly introduces the design components and expected implementation methods of the public opinion monitoring system under the cloud platform.

2.1.1 Design Goals

The public opinion monitoring system under the cloud platform utilizes web crawlers, data analysis and visualization technologies, and the MVC-structured Web platform; to achieve periodic acquisition of hotspot information on news platforms such as Zhihu, Weibo, and Baidu hotspot, and for specific information specified by the user, periodically grab relevant news on various social platforms and news platforms, and after a period of time to visualize the accumulated data, it has shown a certain trend. Public opinion monitoring requires not only breadth, accuracy, and timeliness, so the public opinion monitoring system can not only quickly and timely obtain public opinion hotspot information on multiple platforms, and to ensure the breadth of monitoring public opinion information. According to the needs of users, specific precision information can be captured on multiple platforms to ensure the accuracy of the information; according to the needs of users and the refresh rate of the information platform, different information acquisition cycles can be set to ensure the timeliness of the information [30-39].

2.1.2 System Outline Structure

The implementation of the public opinion monitoring system under the cloud platform is mainly divided into three parts, one is the acquisition and persistence of public opinion information, the second is the analysis and visualization of specific data that the user is concerned about, and the third is the user interaction interface, which mainly includes user login, Information display, interactive data and other modules.

For the acquisition and persistence of the first part of public opinion information, periodically update public opinion hotspot information. For specific information that users care about, you can also customize the real-time multi-platform information acquisition, you can crawl and store on multiple platforms through the Python crawler and the database for implementation. In order to improve efficiency, you can use Python's multi-process technology to achieve multi-task synchronization.

The second is the analysis and visualization of the specific data that users care about. It is mainly implemented through several data analysis and data visualization modules in Python. The public opinion information is no longer only presented through text and links. Graphical to show the hot trends and

proportions that cannot be expressed by the text.

The third is the user interaction interface, which mainly includes several modules such as user login, information display, and interactive data. The front-end page is realized by using HTML+JS+CSS, etc., and the back-end logic function is implemented using PHP. The MVC feature makes the development, debugging and subsequent upgrade of software systems more organized.

These functions can be deployed on the cloud server, enabling users to log in through the browser, set the characteristic information of interest, and monitor the real-time changes of public opinion hot spots and specific attention information. Xia., et al extend the theory of granular computing by covering data space by using granular balls, and the classifier model of non-point input is proposed for the first time [38].

2.2 Feasibility Analysis

Feasibility analysis is to explore and predict a project from multiple angles, analyze whether the project is feasible and has the value of the next step, in order to lay the foundation for the follow-up work. For the public opinion monitoring system under the cloud platform, we mainly start from the three aspects of economy, technology and society to explore whether it is economically feasible for users, the feasibility of technical solutions and the feasibility of social level. From a multi-granularity point of view, a novel exact k-means efficient algorithm is proposed. Especially on the challenging large-k problem, it can exceed all other k-means algorithms [39].

2.2.1 Economic Feasibility

For small and medium-sized organizations and individual users, the public opinion monitoring system under the cloud platform will save a lot of server operation and maintenance costs because the main processing system is deployed on the cloud server. And because of the friendly user interaction design, the use cost and learning cost of the software are greatly reduced; Compared with the survey and warning of human public opinion, non-key monitoring of public opinion information, the public opinion monitoring system can obtain public opinion information on multiple platforms in a timely manner; In the monitoring of user-specified public opinion information, you can also personally specify the refresh time and timely warning. The resulting benefits can completely cover the cost of setting up the system, and as the time of use increases, the resulting long-term benefits and the loss of public opinion avoided by timely warning will increase.

2.2.2 Technical Feasibility

In the public opinion monitoring system under the

cloud platform, the main application technologies are: Python-based web crawlers, data analysis and visualization, MVC-based ThinkPHP5.0 framework, LAMP software services, etc.; from a technical perspective, because all mature technologies are used, so the development of this system is completely achievable.

As a language with low learning cost and a large number of open source libraries, Python has great advantages in web crawlers and data analysis and visualization, and the difficulty of implementing technology is low. For example, there are mature request and re libraries for data acquisition, matching and extraction, matplotlib library for data visualization, etc. JetBrains PyCharm Community Edition 2019.2.5 is used in the development of relevant parts of python.

The ThinkPHP5.0 framework is an open source MVC framework that can be flexibly changed according to its own needs. It has mature routing processing, supports composer for perfect package management, and is very friendly to the use of mysql database, exception handling, models, template engines, etc.; Use Zend Studio to develop in PHP related development process.

The main development environment is Python3, Wampserver32 integrated PHP5.6.25, Apache2.4.23, MySql5.7.14, mysql management software is phpMyadmin.

2.2.3 Social Feasibility

The timeliness of modern social information is becoming more and more important, and there are more and more social and media platforms. It is increasingly difficult for personal energy to obtain effective information in a timely manner. Therefore, a multi-platform public opinion hotspot aggregation site will greatly save individuals and Cost of time and energy for small and medium organizations; but the enthusiasm of the whole society cannot fully satisfy individual needs, it is difficult to get timely attention to some specific information, and the importance of the public opinion monitoring system is self-evident.

In the process of obtaining public opinion information, we mainly applied crawler technology. According to some requirements on the automatic access to network information involved in the "Data Security Management Measures (Draft for Comment)" issued by the National Internet Information Office of China, Among them, Article 16 contains the use of automated means to collect website data, which must not hinder the normal operation of the website. When the number of visits exceeds one-third of the average daily traffic of the website, the automatic collection should be stopped as required by the website; therefore, the public opinion monitoring system should be The time density of information acquisition is limited to ensure that there is no hindrance to the normal

operation of social media platforms [40-45].

2.3 Summary

This chapter mainly conducts two parts of the system's demand analysis and feasibility analysis, plans the specific realization of the system, and explores the feasibility of the system from the economic, technical and social aspects, and draws The conclusion has certain guiding significance for the design of the system; next we will further explore the design and implementation of the system.

3 System Design

System design is mainly based on the analysis of demand analysis, feasibility analysis and other processes to carry out scientific system functions, technical design, and specific implementation methods; we mainly proceed from module division design, database design, and system security design.

3.1 Module Division Design

According to the development process, our system is mainly composed of the following modules: public opinion monitoring module, user interaction module, environment construction and deployment module.

The public opinion monitoring module is responsible for acquiring and processing corresponding public opinion information on various network platforms according to user needs; it mainly consists of two parts: range monitoring and precise monitoring; range monitoring is for socializing such as Zhihu, Weibo, Baidu hotspot, etc. The media platform monitors, periodically obtains public opinion hotspot information, and stores it in the database for data persistence; accurate monitoring is based on the demand obtained during the interaction with the user, and accurately obtains the demand information on the corresponding social platform. After that, the acquired information is stored in the database, and the data visualization module in Python is used to visualize the data and store the visualization results.

The user interaction module is mainly responsible for interacting with users, obtaining information, and displaying them in categories; there are four main components: registration/login, information display main page, user-specific information setting page and user-specific information display page; The registration/login module needs to design database user attribute values, design user login state retention methods, use cookies or sessions to save user login state, and interact with the database; The information display main page needs to extract and display the platform public opinion information stored in the database. For the layout of the display page, the analysis of whether the user is logged in to display different pages; The user-specific information setting

page needs to obtain and store the information set by the user through the Post or Get method, and also involves the layout of the page and the design of the database setting information; The user characteristic information display page needs to display the acquired and processed data in the database, which involves the layout of the display page and the monitoring of the user status.

The environment construction and deployment module is mainly responsible for the construction of the development environment during the development of the software system and the deployment link after the completion of software development. The construction of the development environment is mainly developed under the windows10 operating system, and the two IDEs of PyCharm and Zend Studio are used for development. The environments are Python3 and Wampserver32. The deployment environment is mainly built on the linux server, using the Fedora Linux distribution to operate, deploy PHP, Apache, MySQL software services, and finally build the software system on this set of service platforms to achieve normal operation.

The overall module design of the system is shown in Figure 1.

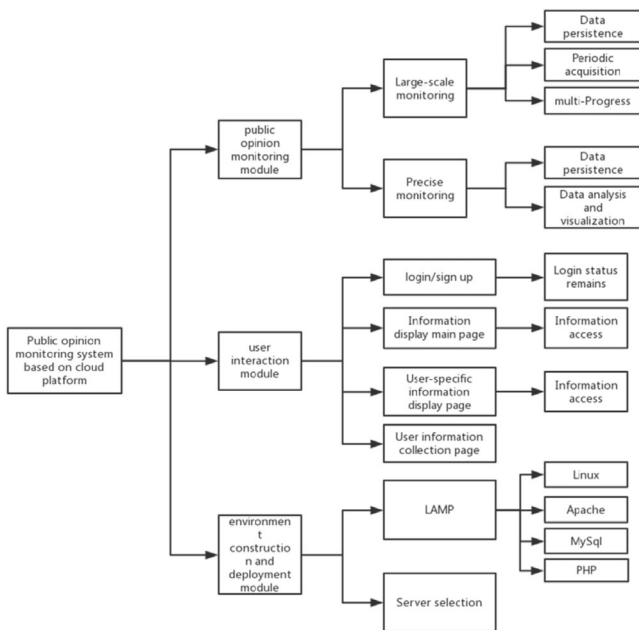


Figure 1. Design of system module

3.2 Database Design

The database is an extremely important part of a software system, which is mainly responsible for data persistence. A good database design can improve the operating efficiency of the software system, save resources, and reduce the difficulty of development. The public opinion monitoring system under the cloud platform uses the mysql database, the version is 5.7.14; we mainly proceed from two parts: conceptual model design and database specific design.

3.2.1 Database Conceptual Model Design

The public opinion monitoring system database under the cloud platform mainly includes two parts: the public opinion database and the user database; the public opinion database is mainly divided into two types: the user-specified public opinion information database and the platform public opinion hotspot database; the user database is mainly for the user information database and the user set public opinion information database.

The ER diagrams of these two types of databases are shown in Figure 2, Figure 3, Figure 4, and Figure 5:

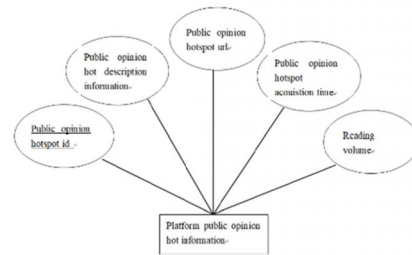


Figure 2. ER diagram of hotspot information

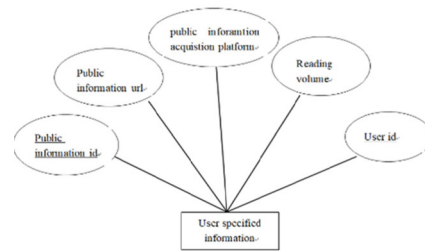


Figure 3. ER diagram of user specified information

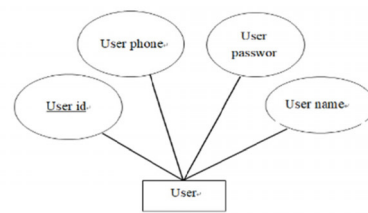


Figure 4. ER diagram of user info

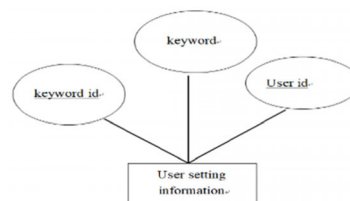


Figure 5. ER diagram of user setting info

3.2.2 Database Specific Design

The database of this system is named “webcrawler”, and the specific design of the main tables is as follows:

1. User information table, the table name is “user_info”, Table 1, which mainly stores user’s basic information:

2. User setting information table, the table name is userkeywords, Table 2, which mainly stores the specific public opinion keywords of the user’s settings:

3. Platform public opinion hotspot information table, Table 3, which mainly stores real-time public opinion

hotspot information obtained from various platforms, Takingzhihu platform hotspot information as an example, the table name is zhihuhotquestions:

4. The specific public opinion information table, Table 4, mainly stores the public opinion hotspot information that the user specifically sets the attention, and the table name is userMessage:

Table 1. User information table

Field name	Description	Type of data	Length	Can it be empty	Whether primary key	Auto increment
User_id	User id	mediumint	9	No	Yes	Yes
User_name	User name	varchar	32	No	No	No
User_phone	User account	varchar	11	No	No	No
User_password	User password	varchar	64	No	No	No

Table 2. User setting information table

Field name	Description	Type of data	Length	Can it be empty	Whether primary key	Auto increment
Message_id	Message id	mediumint	9	No	Yes	Yes
Message_descr	Information description	varchar	32	No	No	No
User_id	User id	mediumint	9	No	No	No

Table 3. Platform hotspot information table

Field name	Description	Type of data	Length	Can it be empty	Whether primary key	Auto increment
questionsId	Zhihu not issue ID	mediumint	9	No	Yes	Yes
questionDeScr	Zhihu not issue description	varchar	255	No	No	No
QuestionURL	Zhihu not issue URL	varchar	255	No	No	No
questionsTime	acquisition time	Timestamp		No	No	No
questionReadCount	Read volume	varchar	30	No	No	No

Table 4. The specific public opinion information table

Field name	Description	Type of data	Length	Can it be empty	Whether primary key	Auto increment
userMessageId	User setting message id	mediumint	9	No	Yes	Yes
UserMessageURL	User setting message URL	varchar	255	No	No	No
Platform	Acquisition platform	varchar	255	No	No	No
readCount	Read volume	Varchar	32	No	No	No
User_id	User id	mediumint	9	No	No	No
getTime	Acquisition time	timestamp				

3.3 System Security Design

The security design in the software system is to ensure that the software system will not be manipulated by humans during operation to cause information leakage, operation failure, performance degradation, and poor user experience.

Therefore, we need to ensure the security of the system from the following aspects:

1. Data Security
- 2, System security
3. Network Security
4. Management Security

Data Security, the data of our entire system is mainly stored in the database. For the safety of the data, it is a better choice to encrypt and store the data. For example, the user’s password can be encrypted with MD5, otherwise the plaintext storage may cause In the

case of database leakage, user information will be clear at a glance; database passwords should also be changed regularly to prevent leakage, or database attacks will lead to database information leakage; establish a good disaster recovery system, regularly backup the database to save content To improve the fault tolerance rate in case of database problems.

System security, It is necessary to set some standards or rules in the system development process, for example, strictly control user rights, prevent malicious users from entering the management background, and maliciously use management rights; perform post pass value for important information instead of get pass value, to prevent some information from being displayed in In the URL; in the system design, flexible use of address routing to prevent the directory information on the server side from being leaked; the legitimacy of user input information must

be judged to prevent SQL injection attacks.

Network Security, Set up the system's network protection wall; limit the access frequency of the ip address to prevent being attacked by DOS. It is best to encrypt the settings in the cookie setting to prevent some data in the cookie information from being leaked; use it under conditional conditions HTTPS protocol, encrypted access.

Management Security, Developers and users must have strict authority management to prevent internal loopholes; regularly change the background management password; remind users and managers that the password format is complex, and simple passwords cannot be used.

The security design of the software system cannot only rely on strict protection in a certain area, but it needs to be fully understood and improve the security awareness of users and management personnel, so as to ensure that the system is as safe as possible.

3.4 Summary

This chapter focuses on the system design of the public opinion monitoring system under the cloud platform. It mainly divides the design of the module, determines the technology and architecture used, the design method of each module, implements the function; database design, determines the database architecture and table allocation; system Safety design, to ensure the safety of the system from multiple angles. In the next chapter, we talk about specific implementation based on the design structure of this chapter.

4 System Implementation and Testing

After the system requirements analysis and detailed design stage, we have clear goals and understanding of the system's specific requirements and implementation methods. In the system implementation and testing phase, we mainly focus on the system's technical solutions, development environment, and module specific implementation methods. And test design and

Table 5. The specific software version table

Development environment operating system	Windows 10 18363.836
Crawler module development language	Python 3.7.1
Crawler module development tool	Pycharm
User interaction module development language	PHP 5.6.25
User interaction module development tools	Zend Studio
User server	Aphace 2.4.23
Database	MySql 5.7.14
Server operating system	Fedora 29

4.3 System Module Implementation

4.3.1 User Interaction Module

The user interaction module is mainly divided into the following parts: user login module, platform public

execution.

4.1 Technical Solutions

Based on the public opinion monitoring system under the cloud platform, using server cloudization to save user costs, using open source projects for personalized customization; using python's mature third-party library for data acquisition, analysis, and visualization, And use the mature Web service software such as MySql, PHP, Linux, Aphace for user management and information output.

The network topology diagram of the system is shown in Figure 6:

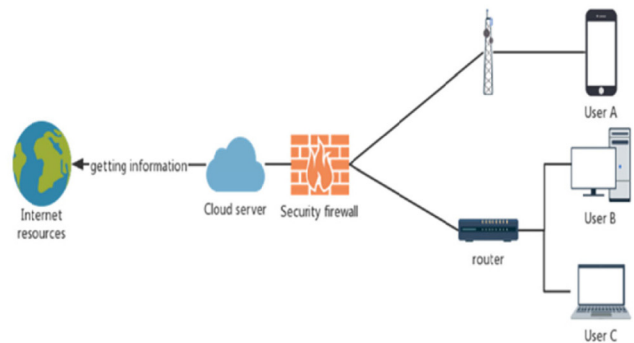


Figure 6. Network topology diagram of the system

4.2 Development Environment

The public opinion monitoring system based on the cloud platform is mainly applied to Python crawler technology, PHP, etc. The database uses the relational database MySql, and many modules in the software system use B/S structure. The development of the user interaction module uses MVC-based ThinkPHP5.0 framework; the system can run on the cloud server where the LAMP software service is deployed. LAMP is open source software, so the compatibility is extremely high.

The specific software version is shown in the following Table 5:

opinion hotspot display module, user attention specific information display module, user specific information setting module; implemented using ThinkPHP5.0 framework and mydql database.

The user interaction logic diagram is as follows, Figure 7:

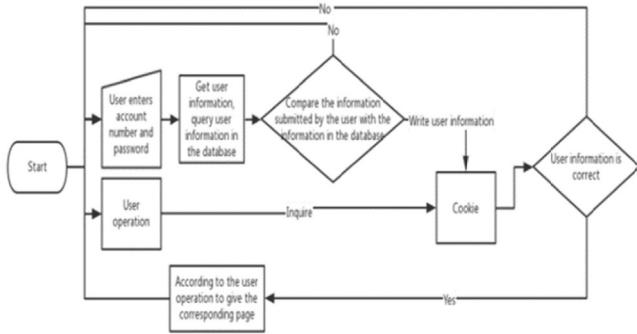


Figure 7. Schematic diagram of user interaction

The user login module is mainly to provide users with login pages and user rights management. The login status is obtained by comparing the user’s account and password with the information stored in the database, and using cookies to maintain the user’s login status for a certain period of time, through inheritance. The base class to obtain permissions, based on the information stored in the cookie to determine the user and the permissions they have.

The realization of the two parts of the platform public opinion hotspot display module and the user attention specific information display module is basically the same. We take the platform public opinion hotspot display module as an example to make an introduction. In the ThinkPHP5.0 framework, users enter from the public entry file and in the application file There are controller, model, and view sections, namely controller, model, and view. The controller calls different models to access data from the database according to the operation generated by the user, and then calls the corresponding view page to display to the user, to achieve a complete operating logic.

4.3.2 Information Acquisition Module

The information acquisition module is mainly divided into platform information acquisition and specific information acquisition. It is implemented in Python language, which uses requests (responsible for obtaining network resources), re (responsible for regular matching information), pymysql (processing database), time (Time related library functions), selenium (through webdriver simulation operation to obtain network resources), bs4 (through beautifulsoup processing to obtain valid information) these several modules.

Platform information acquisition; first obtain network resources through the requests library, extract html page resources among them, match valid information through regular expressions, use find_all to obtain all valid information and store it in a collection, and then use the obtained data to perform Organize, classify, and store valid data in the database, which is shown in Figure 8.

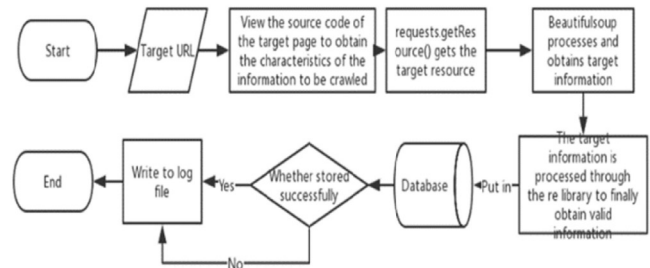


Figure 8. Flow chart of platform information acquisition

For the specific code implementation part, we take the acquisition of knowing popular problems as an example. As follows:

```
def zhihu_web_crawler():
    conn = connect_mysql()
    //Get a database connection
    f = open('G://web_crawler_log/zhihuHotNews.txt', 'a+')
    //Get log file handle
    seedUrl = 'https://www.zhihu.com/signin?next=%2F#signin'
    headers = {'User-Agent': '.....'}
    cookies = {'cookie': '.....'}
    //Set the target link and header and cookie information. Some websites need to log in to obtain valid information.
    Therefore, you need to set the header and cookie information to obtain trust
    soup = getResource (seedUrl, headers, 'utf-8', cookies)
    //Get resource from target URL
    regexFir = re.compile (r'HotItem-content')
    regexSec = re.compile (r'^https://www\.zhihu\.com/question/')
    try:
        with conn.cursor() as cursor:
            for divResource in soup.find_all ('div', attrs={'class': regexFir}):
                //Obtain target information from processed resources
                divResource = str(divResource)
                soup2 = BeautifulSoup(divResource, 'html.parser')
```

```

hotReadCount = re.search('</svg>.+<span c', divResource)
hotReadCount = hotReadCount.group(0)
hotReadCount = str(hotReadCount)
hotReadCount = hotReadCount[6:-7]
for tags in soup2.find_all('a', attrs={'href': regexSec}):
    if 'href' in tags.attrs:
        urlResource = tags['href']
        tags = str(tags)
        qresult = re.search('le">.+</h2', tags)
        qresult = qresult.group(0)
        qresult = str(qresult)
        qresult = qresult[4:-4]
        //Process the acquired target information to obtain valid information
        nowTime = time.time()
        nowTime = str(nowTime)
        result = cursor.execute(
            '.....',
            'values(%s,%s,%s,NOW())',
            (qresult, urlResource, hotReadCount)
        )
        //Store valid information in the database
    )
    if result:
        message = 'Success insert: urlResource:' +
            '.....\n'
        f.write(message)
    else:
        message = 'Failed insert: urlResource:' +
            '.....\n'
        f.write(message)
    //Write stored results to log file
finally:
    f.close()
    conn.close()
//Close the log file, disconnect the database link

```

Obtain specific information; mainly through webdriver to simulate user operation of the browser. First set the target URL, open the target page, locate the control that needs to be operated, write information or other operations to the control, the browser jumps to the target On the information page, next obtain the target page resources, process the resources, the regular expression matches the target information, process the target information to obtain valid information, and store the valid information in the database.

The implementation process is shown in the Figure 9:

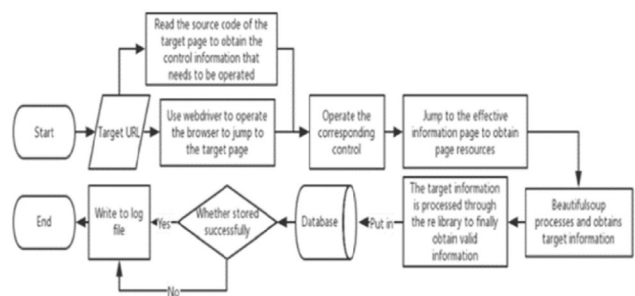


Figure 9. Schematic diagram

For the specific code implementation part, we take the Weibo keyword information acquisition as an example, as shown below:

```

def weibo_webdriver():
    f = open('G://weibo_webdriver_log.txt', 'a+')
    driver = webdriver.Chrome()
    driver.get('https://weibo.com')
    time.sleep(10) //Delay the execution of the following code to delay the page loading for a period of 10 seconds
    inputElement = driver.find_element_by_css_selector('.....') //Get search controls
    searchButton = driver.find_element_by_css_selector('.....') //Get search button control

```

```

input tElement.send_keys('.....') //Write value in search input control
searchButton.click()
time.sleep(5) //Get the HTML resources after the code is executed
html = driver.page_source //Get html resources
soup = BeautifulSoup(html, 'html.parser')
for tag in soup.find_all('div', attrs={'class': 'card-topic-a'}):
// Get the number of reads and posts and write to the file
    result1 = re.search('<span>.+<span>', tag)
    result1 = str(result1.group(0))
    result1 = result1[6, -13]
    result1 = 'read volume:' + result1 + ' '
    f.write(result1)
    result2 = re.search('</span>.+</span>', tag)
    result2 = str(result2.group(0))
    result2 = result2[13, -7]
    result2 = 'Posts' + result2 + ' \n'
    f.write(result2)
for tag in soup.find_all('div', attrs={'class': 'content'}):
//Get specific content and write to file
    username_result = re.search('nick-name=.\+\>\.\.\.', tag)
    username_result = str(username_result.group(0))
    username_result = "username:" + username_result[10, -4] + '\n'
    like_result = re.search('<em>.+</em>', tag)
    like_result = str(like_result.group(0))
    like_result = "Likes" + like_result[4, -5] + '\n'
    tag = BeautifulSoup(str(tag), 'html.parser')
    txt_result = str(soup.find('p', attrs={'class': 'txt'})) + '\n'
    f.write(username_result + like_result + txt_result)
f.close() //Close the log file

```

4.4 System Test

4.4.1 Test Plan

The public opinion monitoring system test under the cloud platform is divided into two parts, information acquisition and user interaction;

Test target: crawler test, mainly need to test the reliability of the system, whether it can obtain the platform information normally, and carry out persistence; user interaction interface test, mainly need to test whether the interface jumps normally and whether the user authority is complete.

Test environment is shown in Table 6.

Table 6. Test environment

operating system	Windows 10
Python Operating environment	Python 3
Web Software service environment	Wampserver 32
Information acquisition test browser	Google Chrome (Kernel version 75)
User interaction test browser	Google Chrome (Kernel version 75)

4.4.2 Test Platform Construction

Information acquisition module environment

construction: download the Python installation package, run the installation package, enter the Python installation wizard, and complete the installation using the default configuration.

User interaction module environment construction: Wampserver32 installation, download the installation package, enter the installation wizard, complete the installation using the default configuration, modify the server port to 8080 in the php.ini file, add content in the httpd.config file, so that it can access the tp5 project.

Browser installation: Uninstall the local Chrome browser, turn off the Google automatic upgrade service, and install the Google Chrome browser with a kernel version of 75.

4.4.3 Test Process and Results

Set the information acquisition module to run automatically at regular intervals, view the operation results, the software runs successfully, and successfully obtains information. We intercepted the database diagram of the information acquisition module as an example. The database information after the information acquisition module runs is as follows, Figure 10:

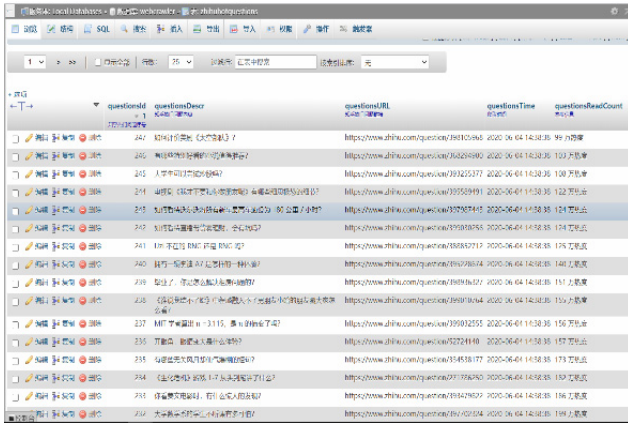


Figure 10. Screenshot of the running database

The user interaction module mainly has the following pages: user landing page, platform information display page, and user specific information display page. After running, the following results are obtained in Figure 11 to Figure 13:

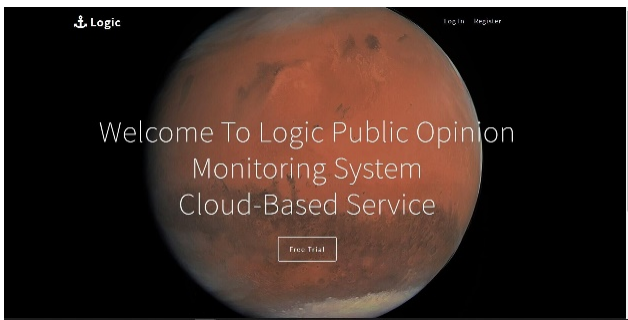


Figure 11. Screenshot of the main page of the system

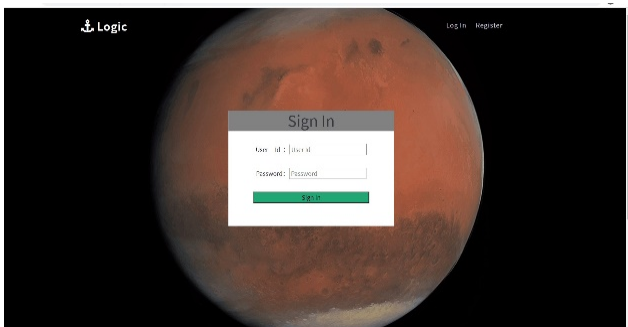


Figure 12. Screenshot of user login interface



Figure 13. Screenshot of platform information display

The testing process mainly involves access to various user interfaces; testing operations that users need to submit information, such as login, user-specific information settings, etc.; Make some incorrect entries in the input box to see if there are good tips for the user; Whether the functional page is displayed normally; Whether the jump between the corresponding links is normal.

After a comprehensive test of the functions of the user interaction interface, after positioning and modifying some of the functions that failed link jumps, and after passing the comprehensive test again, the interface is basically complete.

4.5 Summary

In this chapter, we mainly introduce the implementation and testing of the system, and clarify our technical solutions and development environment; in the system implementation module, the implementation method and key code of each module are shown; in the system testing stage, We have specifically determined the test plan and the software of the test environment, determined the expected goals, executed the test process, and displayed the test results; next, we will make a complete summary of this article.

5 Conclusion

In the modern society where information platforms are developing faster and more and more, the hot spots of public opinion are likely to be fleeting. Too many sources of information make it difficult for individuals and small and medium-sized organizations to quickly, accurately and timely obtain information on various platforms. Information flow Interfering with our judgment; therefore, the multi-platform integrated public opinion monitoring system can meet the needs of some people; however, in fact, we cannot just pay attention to the public opinion hotspots, and for some special information, individuals will also have concerns. Therefore, this project has added monitoring and storage for specific information social and media platforms.

Public opinion monitoring system as a method to replace manual network information integration, firstly saves labor costs, greatly reduces the user's cost, monitoring costs will mainly cover software development and subsequent upgrades; second, compared with manual inefficiency and The disadvantages of not being able to work around the clock, the public opinion monitoring system is well-designed and efficiently implemented by the computer, which not only greatly improves the efficiency, but also can be monitored and collected around the clock.

We chose the B/S structure as the basic structure of the project. On the one hand, considering the difficulty of software development will be reduced, eliminating

the need for client development; on the other hand, considering the convenience and user-friendliness of subsequent software upgrades; Compared with the C/S structure, the B/S structure only needs to be updated on the server side to realize the changes in project requirements, and users quickly feel the software upgrade.

Finally, this article mainly focuses on the development process and technical exploration of software systems. For the expected goals: platform information collection and user interaction system design are also basically achieved. Of course, the software system also has shortcomings, compared with large-scale The personalized and customized service of the software company also has the goal of further improvement in certain functions and details; the text matching, data persistence, and structural design of the system can be further optimized and improved; in terms of system security It can also be further improved in terms of storage of some data and restrictions on user rights.

In short, the completion of this article is more focused on the basic implementation of the system. Due to empirical reasons, it is not particularly perfect from a technical point of view, so there is further work to be done to improve the system.

Acknowledgements

This job is supported by the National key R&D Program of China under Grant NO. 2018YFB0203901 and the Key Research and Development Program of Shaanxi Province (No. 2018ZDXM-GY-036) and Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data (No. IPBED7).

References

- [1] T. Teng, M. Lefley, D. Claremont, Progress Towards Automated Diabetic Ocular Screening: A Review of Image Analysis and Intelligent Systems for Diabetic Retinopathy, *Medical and Biological Engineering and Computing*, Vol. 40, No. 1, pp. 2-13, January, 2002.
- [2] C. Köse, C. Ikibas, A Personal Identification System Using Retinal Vasculature in Retinal Fundus Images, *Expert Systems with Applications*, Vol. 38, No. 11, pp. 13670-13681, October, 2011.
- [3] M. Kaur, R. Talwar, Review on: Blood Vessel Extraction and Eye Retinopathy Detection, *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 6, pp. 7513-7516, December, 2014.
- [4] T. Mapayi, S. Viriri, J.-R. Tapamo, Comparative Study of Retinal Vessel Segmentation Based on Global Thresholding Techniques, *Computational and Mathematical Methods in Medicine*, Vol. 2015, pp. 1-15, February, 2015. doi: 10.1155/2015/895267.
- [5] F. Bukenya, L. Bai, A. Kiweewa, A Review of Blood Vessel Segmentation Techniques, *In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, 2018, pp. 1-10. doi: 10.1109/cais.2018.8441989.
- [6] M. Nergiz, M. Akın, Retinal Vessel Segmentation via Structure Tensor Coloring and Anisotropy Enhancement, *Symmetry*, Vol. 9, No. 11, Article No. 276, November, 2017.
- [7] A. M. Mendonca, A. Campilho, Segmentation of Retinal Blood Vessels by Combining the Detection of Centerlines and Morphological Reconstruction, *IEEE Transactions on Medical Imaging*, Vol. 25, No. 9, pp. 1200-1213, September, 2006. doi: 10.1109/tmi.2006.879955.
- [8] M. A. Palomera-Perez, M. E. Martinez-Perez, H. Benitez-Perez, J. L. Ortega-Arjona, Parallel Multiscale Feature Extraction and Region Growing: Application in Retinal Blood Vessel Detection, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 14, No. 2, pp. 500-506, March, 2010. doi: 10.1109/titb.2009.2036604.
- [9] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, S. A. Barman, Blood Vessel Segmentation Methodologies in Retinal Images – A Survey, *Computer Methods and Programs in Biomedicine*, Vol. 108, No. 1, pp. 407-433, October, 2012. doi: 10.1016/j.cmpb.2012.03.009.
- [10] Z. Xiao, M. Adel, S. Bourennane, Bayesian Method with Spatial Constraint for Retinal Vessel Segmentation, *Computational and Mathematical Methods in Medicine*, Vol. 2013, pp. 1-9, July, 2013. doi: 10.1155/2013/401413.
- [11] S. Manoj, Muralidharan, P. M. Sandeep, Neural Network Based Classifier for Retinal Blood Vessel Segmentation, *International Journal of Recent Trends in Electrical & Electronics Engineering*, Vol. 3, No. 1, pp. 44-53, December, 2013.
- [12] D. Marin, A. Aquino, M. E. Gegundez-Arias, J. M. Bravo, A New Supervised Method for Blood Vessel Segmentation in Retinal Images by Using Gray-Level and Moment Invariants-Based Features, *IEEE Transactions on Medical Imaging*, Vol. 30, No. 1, pp. 146-158, January, 2011. doi: 10.1109/tmi.2010.2064333.
- [13] A. Budai, R. Bock, A. Maier, J. Hornegger, G. Michelson, Robust Vessel Segmentation in Fundus Images, *International Journal of Biomedical Imaging*, Vol. 2013, pp. 1-11, December, 2013. doi: 10.1155/2013/154860.
- [14] W. Wei, Y. Qi, Information Potential Fields Navigation in Wireless Ad-Hoc Sensor Networks, *Sensors*, Vol. 11, No. 5, pp. 4794-4807, May, 2011.
- [15] W. Wei, M. Wozniak, R. Damasevicius, X. Fan, Y. Li, Algorithm Research of Known-plaintext Attack on Double Random Phase Mask Based on WSNs, *Journal of Internet Technology*, Vol. 20, No. 1, pp. 39-48, January, 2019.
- [16] W. Wei, X. Fan, H. Song, H. Wang, Video Tamper Detection Based on Multi-scale Mutual Information, *Multimedia Tools & Applications*, Vol. 78, No. 19, pp. 27109-27126, October, 2019. doi: 10.1007/s11042-017-5083-1.
- [17] X. Fan, M. Wozniak, H. Song, W. Li, Y. Li, P. Shen, H ∞

- Control of Network Control System for Singular Plant, *Information Technology And Control*, Vol. 47, No. 1, pp. 140-150, March, 2018.
- [18] W. Wei, H. M. Srivastava, Y. Zhang, L. Wang, P. Shen, J. Zhang, A Local Fractional Integral Inequality on Fractal Space Analogous to Anderson's Inequality, *Abstract and Applied Analysis*, Vol. 2014, Article ID 797561, June, 2014.
- [19] W. Wei, X. L. Yang, P. Y. Shen, B. Zhou, Holes Detection in Anisotropic Sensornets: Topological Methods, *International Journal of Distributed Sensor Networks*, Vol. 8, No. 10, Article ID 135054, October, 2012.
- [20] W. Wei, X. L. Yang, B. Zhou, J. Feng, P. Y. Shen, Combined Energy Minimization for Image Reconstruction from FEw Views, *Mathematical Problems in Engineering*, Vol. 2012, Article ID 154630, October, 2012. doi: 10.1155/2012/154630.
- [21] W. Wei, Y. Qiang, J. Zhang, A Bijection between Lattice-Valued Filters and Lattice-Valued Congruences in Residuated Lattices, *Mathematical Problems in Engineering*, Vol. 2013, Article ID 908623, July, 2013.
- [22] W. Wei, Z. Sun, H. Song, H. Wang, X. Fan, X. Chen, Energy Balance-Based Steerable Arguments Coverage Method in WSNs, *IEEE Access*, Vol. 6, pp. 33766-33773, March, 2017. doi: 10.1109/ACCESS.2017.2682845.
- [23] W. Wei, J. Su, H. Song, H. Wang, X. Fan, Cdma-based Anti-Collision Algorithm for Epc Global c1 gen2 Systems, *Telecommunication Systems*, Vol. 67, No. 3, pp. 63-71, January, 2018.
- [24] W. Wei, X. Fan, H. Song, X. Fan, J. Yang, Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud Computing, *IEEE Transactions on Services Computing*, Vol. 11, No. 1, pp. 78-89, January-February, 2018.
- [25] W. Wei, B. Zhou, D. Polap, M. Wozniak, A Regional Adaptive Variational PDE Model for Computed Tomography Image Reconstruction, *Pattern Recognition*, Vol. 92, pp. 64-81, August, 2019. doi: 10.1016/j.patcog.2019.03.009.
- [26] W. Wei, X. Xia, M. Wozniak, X. Fan, R. Damasevicius, Y. Li, Multi-sink Distributed Power Control Algorithm for Cyber-physical-systems in Coal Mine Tunnels, *Computer Networks*, Vol. 161, pp. 210-219, October, 2019. doi: 10.1016/j.comnet.2019.04.017.
- [27] W. Wei, H. Song, W. Li, P. Shen, A. Vasilakos, Gradient-Driven Parking Navigation Using a Continuous Information Potential Field Based on Wireless Sensor Network, *Information Sciences*, Vol. 408, pp. 100-114, October, 2017.
- [28] W. Wei, Q. Xu, L. Wang, X. H. Hei, P. Shen, W. Shi, L. Shan, GI/Geom/1 Queue Based on Communication Model for Mesh Networks, *International Journal of Communication Systems*, Vol. 27, No. 11, pp. 3013-3029, November, 2014.
- [29] Q. Ke, J. Zhang, H. Song, Y. Wan, Big Data Analytics Enabled by Feature Extraction Based on Partial Independence, *Neurocomputing*, Vol. 288, pp. 3-10, May, 2018.
- [30] Q. Ke, J. Zhang, W. Wei, R. Damasevicius, M. Wozniak, Adaptive Independent Subspace Analysis of Brain Magnetic Resonance Imaging Data, *IEEE Access*, Vol. 7, pp. 12252-12261, January, 2019.
- [31] Q. Ke, J. Zhang, M. Wozniak, W. Wei, The Phase and Shift-Invariant Feature by Adaptive Independent Subspace Analysis for Cortical Complex Cells, *Information Technology And Control*, Vol. 48, No. 1, pp. 58-70, March, 2019.
- [32] X. Wang, Y. Qi, Z. Wang, Y. Chen, Y. Zhou, Design and Implementation of SecPod: A Framework for Virtualization-Based Security Systems, *IEEE Transactions on Dependable and Secure Computing*, Vol. 16, No. 1, pp. 44-57, January-February, 2019.
- [33] S. Qi, Y. Zheng, X. Chen, W. Wei, Ants can Carry Cheese: Secure and Private RFID-Enabled Third-Party Distribution, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-14, September 2020. doi: 10.1109/TDSC.2020.3026191.
- [34] S. Qi, Y. Lu, W. Wei, X. Chen, Efficient Data Access Control with Fine-Grained Data Protection in Cloud-Assisted IIoT, *IEEE Internet of Things Journal*, Vol. 8, No. 4, pp. 2886-2899, February, 2021. doi: 10.1109/JIOT.2020.3020979.
- [35] Q. Ke, J. Zhang, W. Wei, D. Polap, M. Wozniak, L. Kosmider, R. Damasevicius, A Neuro-heuristic Approach for Recognition of Lung Diseases from X-ray images, *Expert Systems with Applications*, Vol. 126, pp. 218-232, July, 2019.
- [36] J. Su, Z. Sheng, L. Xie, G. Li, A. X. Liu, Fast Splitting-based Tag Identification Algorithm for Anti-collision in UHF RFID System, *IEEE Transactions on Communications*, Vol. 67, No. 3, pp. 2527-2538, March, 2019. doi: 10.1109/TCOMM.2018.2884001.
- [37] J. Su, Z. Sheng, V. C. M. Leung, Y. Chen, Energy Efficient Tag Identification Algorithms for RFID: Survey, Motivation and New Design, *IEEE Wireless Communications*, Vol. 26, No. 3, pp. 118-124, June, 2019.
- [38] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, Granular Ball Computing Classifiers for Efficient, Scalable and Robust Learning, *Information Sciences*, Vol. 483, pp. 136-152, May, 2019.
- [39] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, Z. Chen, A Fast Adaptive k-means with No Bounds, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1, July, 2020. doi: 10.1109/TPAMI.2020.3008694.
- [40] J. Su, R. Xu, S. Yu, B. Wang, J. Wang, Idle slots skipped Mechanism Based Tag Identification Algorithm with Enhanced Collision Detection, *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 5, pp. 2294-2309, May, 2020.
- [41] J. Su, R. Xu, S. Yu, B. Wang, J. Wang, Redundant Rule Detection for Software-defined Networking, *KSII Transactions on Internet and Information Systems*, Vol. 14, No. 6, pp. 2735-2751, June, 2020.
- [42] S. Liu, D. Liu, K. Muhammad, W. Ding, Effective Template Update Mechanism in Visual Tracking with Background Clutter, *Neurocomputing*, November, 2020. doi: 10.1016/j.neucom.2019.12.143.
- [43] S. Liu, D. Liu, G. Srivastava, D. Polap, M. Wozniak, Overview and Methods of Correlation Filter Algorithms in Object Tracking, *Complex and Intelligent Systems*, June, 2020. doi: 10.1007/s40747-020-00161-4.

- [44] S. Liu, S. Wang, X. Liu, C. -T. Lin, Z. Lv, Fuzzy Detection Aided Real-Time and Robust Visual Tracking Under Complex Environments, *IEEE Transactions on Fuzzy Systems*, Vol. 29, No. 1, pp. 90-102, January, 2021. doi: 10.1109/TFUZZ.2020.3006520.
- [45] S. Liu, C. Guo, F. Al-Turjman, K. Muhammad, V. H. C. de Albuquerque, Reliability of response region: A novel mechanism in visual tracking by edge computing for IIoT environments, *Mechanical systems and signal processing*, Vol. 138, pp. 106537.1-106537.15, April, 2020.

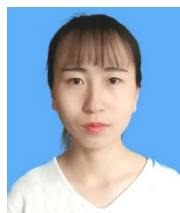
Biographies



Wei Wei is an associate professor of School of Computer Science and Engineering , Xi'an University of Technology, Xi'an 710048, China. He is a senior member of IEEE, CCF. He received his Ph.D. and M.S. degrees from Xian Jiaotong University in 2011 and 2005, respectively. He ran many funded research projects as principal investigator and technical members. His research interest is in the area of wireless networks, wireless sensor networks Application, Image Processing, Mobile Computing, Distributed Computing, and Pervasive Computing, Internet of Things, Sensor Data Clouds, etc. He has published around one hundred research papers in international conferences and journals. He is an editorial board member of FGCS, IEEE Access, AHSWN, IEICE, KSII, etc. He is a TPC member of many conferences and regular reviewer of IEEE TPDS, TVT, TIP, TMC, TWC, JNCA and many other Elsevier journals.



Long Wang, Bachelor, is graduated from Xi'an University of technology Xian in 2020, majoring in Computer Science and Technology. His research interest is in the area of web technology and software testing technology.



Xiaowan Li received a bachelor's degree in management from Lanzhou University of Finance and Economics in 2020. She is currently a postgraduate student at Xi'an University of Technology, in China and will receive her M.S. in 2023. Her research interests are big data and Natural language processing.



Beibei Zhang is graduated from Xi'an Jiaotong university in 2012, he did the work related to system engineering, after 2012, he was doing postdoctoral research at Xi'an Jiaotong university. He are mainly engaged in big data research, NLP and OSN and so on. He published more than 30 high-level international SCI academic papers in related fields.



Rafał Scherer received his MSc degree in computer science from the Czestochowa University of Technology, Poland, in 1997 and his PhD in 2002 from the same university. Currently, he is an associate professor at Czestochowa University of Technology. His present research interests include machine learning and neural networks for image processing, computer system security, prediction and classification.

