Software Defect Fault Intelligent Location and Identification Method Based on Data Mining

Fang Wang [1,*], Sungho Park [2], Cattareeya Suwanasri [3]

[1] Wenhua College, Wuhan, 430074, Hubei, China
[2] College of Maritime Sciences, Korea Maritime & Ocean University, Busan, Korea
[3] King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
wangfang@whc.edu.cn*
* corresponding author

(Received January 18, 2023 Revised February 21, 2023 Accepted February 18, 2023, Available online March 1, 2023)

Abstract

With the advancement of the times, computer technology is also constantly improving, and people's requirements for software functions are also constantly improving, and as software functions become more and more complex, developers are technically limited and teamwork is not tacitly coordinated. And so on, so in the software development process, some errors and problems will inevitably lead to software defects. The purpose of this paper is to study the intelligent location and identification methods of software defects based on data mining. This article first studies the domestic and foreign software defect fault intelligent location technology, analyzes the shortcomings of traditional software defect detection and fault detection, then introduces data mining technology in detail, and finally conducts in-depth research on software defect prediction technology. Through in-depth research on several technologies, it reduces the accidents of software equipment and delays its service life. According to the experiments in this article, the software defect location proposed in this article uses two methods to compare. The first error set is used as a unit to measure the subsequent error set software error location cost. The first error set 1F contains 19 A manually injected error program, and the average positioning cost obtained is 3.75%.

Keywords: Software Defect, Data Mining, Fault Location, Data Algorithm

1. Introduction

With the increasing dependence of modern society on software, the function and complexity of software are getting higher and higher. The improvement of software can effectively improve the quality of people's work and life. As people's requirements for software are getting higher and higher, the complexity and scale of software are getting higher and higher, and once errors occur, the loss to people will be greater and greater [1-3].

Software is a critical component of the modern world, and the reliability and accuracy of software are paramount in ensuring the smooth functioning of various systems. However, software defects and faults are inevitable, and identifying and locating these defects can be a daunting task. Therefore, various software testing techniques have been developed over the years, and one such technique is data mining [2]. Data mining is the process of analyzing large sets of data to discover patterns and establish relationships that can be used to make informed decisions. The Software Defect Fault Intelligent Location and Identification Method Based on Data Mining is one such technique that utilizes data mining to identify and locate software defects and faults.

The Software Defect Fault Intelligent Location and Identification Method Based on Data Mining uses a combination of data mining algorithms and techniques to analyze the software testing data and identify defects and faults. This method includes various steps such as data collection, data preprocessing, data analysis, and model building [4]. During the data collection phase, data from various sources such as code repositories, bug tracking systems, and testing reports are gathered. In the data preprocessing phase, the collected data is cleaned and transformed to eliminate any errors and inconsistencies.

In the data analysis phase, various data mining algorithms such as clustering, classification, and association rule mining are used to identify patterns and relationships in the data. These algorithms help to identify the areas of the software that are more likely to have defects and faults. In the model building phase, the identified patterns and relationships are used to build a predictive model that can identify and locate defects and faults in the software [5].

The Software Defect Fault Intelligent Location and Identification Method Based on Data Mining has several advantages over traditional software testing methods. Firstly, it can analyze large sets of data quickly and accurately, which is essential in detecting defects and faults early in the software development process. Secondly, it can identify hidden patterns and relationships in the data that may not be apparent through manual testing. Finally, it can improve the overall quality of the software by providing developers with valuable insights into the areas of the software that require further testing and refinement.

In the research of software defect fault location and identification based on data mining technology, many scholars have studied it and achieved good results. For example: Chen Jialin, Sun Jun, He Yi, Zhang Jinhua, Yang Shuo, Zhao Shiwen and other authors Suggestions are made. Based on the K-Means algorithm, the user’s vibration, humidity, temperature, magnetic field, grid ripple and other external quality samples are obtained through calculations, and then the number of clusters is calculated according to the attributes or characteristics of the samples. , And then classify the sample into a certain category, so that the various data of the sample can be linked together, and then use the Euclidean formula to calculate the data within the category range [1,4,5]. Two authors, Guo Jiangfeng and Qu Yubin, showed that software defect prediction can predict software defects in advance, which greatly reduces the huge manpower and material resources required for testing, and that optimized testing can better detect the occurrence and occurrence of software faults Location. [6-8].

This paper carefully analyzes and introduces data mining technology and software defects based on data mining-based software defect fault intelligent location and identification methods, and then uses software defect location and prediction algorithms to calculate the location of software defects. Data experiments are used to prove the correctness and feasibility of the research direction of this article [9].

2. Software Defects and Data Mining

2.1. Data Mining Technology

2.1.1 The definition of data mining

Data mining technology is a discipline that includes databases, artificial intelligence, parallel computing, and mathematics, and includes the intersection of statistics and imaging technology. It is extracted from a large number of messy, messy, fuzzy descriptions and random application data [10,11]. Come out, people didn't know before, but it is indeed potentially useful information. The above definition includes the following four concepts: first, there must be a large number of actual and messy data sources; second: the obtained data must meet the needs of customers; third: the discovered knowledge must be easy to understand and easy to understand. Acceptable, easy to use, fourth: the data obtained is interrelated, must have specific conditions and restrictions, and can only be applied to specific fields. We view data as the ore from which knowledge is found, and knowledge is like the metal extracted from it. The data obtained by using data mining technology can be applied to the control process, decision selection, information management and query optimization, and data maintenance itself. Data mining technology means that people have broken away from the simple way of querying data and progressed towards knowledge discovery [4,12-15].

2.1.2 Classification of data mining

Association rule method: used to discover the correlation between features in a large number of data sets, such as "95% of customers buy when you buy product A, then buy product B." This rule means that customers buy some products at the same time. Possibility to purchase other goods at that time [16]. Grouping and sorting method: Grouping is the process of dividing a set of real data obtained into several classes of data objects according to the similarity of the data. Sorting must be a hypothesis in the database, assuming that a specific classification in the database is designed, and then the data is allocated according to the classification. Grouping and sorting are divided according to the size target between the class differences [5].

Intelligent learning method: The intelligent learning method is to let the computer use the cognitive model to simulate human thinking, and then extract the data from the data set. Multi-level data processing: database data generally contains initial detailed information [17]. The information in a data set is transformed from the initial level to a higher level of information set, and this technology is called data aggregation. Concept is the whole process of collecting and processing relevant data information from low-level concepts to a high-level concept-level secretary information collection database [18,19]. Neural network method: It is a very suitable method for data mining. Because it has the advantages of excellent reflectivity, merge processing, self-adaptation, word organization, distributed storage and high fault tolerance.

Typical neural network models are mainly divided into three categories: the first is a generalized neural network model used for classification, prediction and recognition; the other is used for related calculation memory optimization and calculation, thereby feeding back to the neural network model ; The third is a self-organizing mapping method suitable for grouping [6,20]. Decision tree method: It is to find the most informative keyword among a large number of secretaries and create it as the root node, and create different branches on this basis. In the set of branches, repeat the process of creating byte points The oldest decision tree method is the ID3 method proposed by JRKunlan in 1993, which has a wide range of influence in the data mining industry. The better the mining effect, the more people have made improvements based on the ID3 algorithm according to their needs, and created many new decision tree methods. Visualization technology: In order to allow users to analyze the data more clearly, the results of the data are transformed into graphs, images, Process visualization [7,21,22].

2.2. Software Defects

2.2.1 Definition of software defects

Software defects are errors that may cause software failure in the software. Because software defects are unavoidable in the process of software design and development, they are sometimes called BUGs [23-25]. It will cause the software to fail to achieve the expected effect of the customer. In order to better manage the quality of the software installation, we will classify the software defects [8].

Error: A human error that causes a software error. Such errors may occur in various stages of software requirements analysis, software design and coding.

Failure: Refers to the operation failure or the operation result is invalid when the software is running or when it is turned on.

2.2.2 Classification of software defects

The general classification of software defects is based on the characteristics of software defects, which are basically classified by the external manifestations of the defects, the source and the severity of the defects.

Classification based on the external manifestation of software defects: classification is based on the stage and location of the external manifestation of defects. They are generally divided into 6 categories of defects: defects in requirements, design, documentation, algorithms, interfaces, and performance. Algorithmic defects: Generally, it is a data structure error, a control flow error, or other data errors. Interface defects: Generally, there are errors in pictures and text buttons on the system interface. Performance defects: Generally, the performance of the system and software is insufficient to meet the needs of customers, performance problems or affected by system performance. Requirement defects: Basically, they are collected to seek errors or incompleteness. Design defects: Generally, they are defects that appear when designing the system. Document defect: Generally, it is caused by an error in the document [9].

According to the source of software defects, software defects can be classified into three categories: program defects: the function of the program does not conform to the corresponding document description, but the document is correct; document defects: the function of the program is different from the corresponding document description, The program meets the requirements but the document description is incorrect. Imperfect design: Due to design errors, even if the program and documentation are correct, the running result does not match the actual requirements or the program state is not available, it does not meet the actual requirements or the state presented by the program is not available [10].

The severity of defects is classified according to the degree of influence or deviation of software defects in software systems and software development. Specifically, it can be divided into 5 categories: Category 1: Problems that cannot fully meet the system requirements, and the basic functions are not fully completed or endanger software security; Category 2: Cannot meet the needs of the system, or perform basic operations that cannot be replaced by others. But the execution of the method can be solved in other ways (loading or restarting the software is not the solution). Category 4: does not affect the requirements of the system but fully meets the performance of the basic functions, but there are problems that the operator is inconvenient to operate; Category 5: other problems [11].

3. Software Defect Location and Prediction Algorithm

3.1. Detection Methods of Software Defects

3.3.1 Bayesian method

It is a diagnostic method based on probability and statistical calculations. Its basis is the density function, which describes the operating state of the equipment system through the collection of error information, so that faults can be diagnosed and analyzed. The diagnostic reasoning process is divided into two parts: (1) pre-probability evaluation, (2) probability calculation. The operation of equipment is a random process, and the possibility of failure can generally be calculated.

3.3.2 Time series method

Time series analysis is to collect continuous signals at equal intervals to obtain different data sets. The mathematical statistics method for processing and analyzing the above data is called time series analysis method. Time series analysis is divided into two types: time domain analysis and frequency domain analysis. Time domain analysis is to use the process of analyzing the signal to evaluate the randomness and periodicity of the signal, and to obtain the law of the sequence. Frequency field analysis is to analyze and disassemble the signal sequence, decompose the composite signal into multiple simple signals, and perform spectrum analysis on the simple signal sequence [12].

3.3.3 Gray system method

Refers to a system in which part of the information is known but other parts of the information are unknown. The difference between the white system and the gray system is whether there is a specific relationship between the various factors of the system. The system factors in the white system have a clear correlation, while the direct relationship of the factors in the gray system is not very clear. Grey system theory is an extension of viewpoints and methods. From a systematic point of view, it studies the relationship between information based on some logical reasoning and rational knowledge.

The equipment fault monitoring and diagnosis system consists of many factors. If the components of the system are clear, and there is a clear relationship between the various factors, then the system is a white system. If part of the system is known and part of the information is unknown, then it is a gray system.

3.3.4 Fault tree analysis

It is a finalized causal model. First, the failure of the equipment software is considered the first level (top-level event), and then the direct cause of the failure (including environment, materials, human error, etc.). At the same level as the second level (the middle event), associate them with the appropriate events, and associate them with the top-level events through appropriate logic. Then, according to the above method, the reasons for the occurrence of intermediate events are extended to the third level, and finally the most basic reasons (the following facts) are analyzed. The error tree analysis reflects all the logical relationships between the error vector and the error vector.

3.2. Static Defect Prediction

The numerical static software defect prediction technology in this paper uses software metrics to solve the problems of predicting software defects. In the early days, it was mainly through the calculation of the size of the software, and then the prediction of the software defects, that is, by studying the relationship between the software defects and the software size, to predict the possible existence of the software and the number of defects in the software.

In the 1970s, software metrics have been used to establish software defect models. The specific management is:

$$B = 4.86 + 01018p \qquad (1)$$

B is the number of defects that may be included before the test and P is the number of lines of code. Other scholars proposed that the software defect prediction model is based on the volume of the software and the relationship between the number of operands and defects:

$$V = N\, log_2(\eta) \qquad (2)$$

Where V is the volume of the software, $N = N_1 + N_{2,}$ N1 represents the total number of operators, N2 represents the total number of operands,$\eta = \eta_1 + \eta_2$, η_1 is the number of different operators, and η_2 is the number of different operands.

$$B = V/3000 \qquad (3)$$

Among them, B is the number of possible defects. On this basis, the relationship between the number of defects B and the number of executable codes P is improved.

$$\frac{B}{P} = A_0 + A_1 In\, P + A_2 In^2 P \qquad (4)$$

The A is related to the language used by the program.

3.3. Dynamic Fault Defect Detection

Dynamic defect prediction technology is based on the relationship between time and defects. Many dynamic software reliability models are based on dynamic failure prediction technology. Each defect in the software is independent and interrelated. The probability of each software failure causing system failure is the same, and the interval between each failure is also independent of each other. Deal with software defects detected during testing, and only eliminate one defect at a time. The time for handling errors is ignored, and no new software defects are caused when software defects are fixed.

The probability of software failure is a constant at each failure interval, and its value is proportional to the number of remaining defects in the software. In the i-th interval, the failure rate function is: $Z(t_i) = \Phi(B - i = 1)$Among them, Φ is the proportional constant t_i is the time variable starting from the i-th defect interval from the i-1th defect occurrence.

4. Software Defect Testing and Analysis Based on Data Mining

As shown in Table 1, this article chooses 4 medium-sized C language programs A1, A2, A3, and A4 among a certain tool program as experimental objects, and artificially injects errors for each version of the program to generate 1F to 6F and other version errors.

Table. 1. Table of subjects

Error set	A1	A2	A3	A4
1F	19	15	12	23
2F	23	22	21	20
3F	25	22	21	24
4f	25	25	20	21
5f	26	28	19	13
6f	25	27	16	11

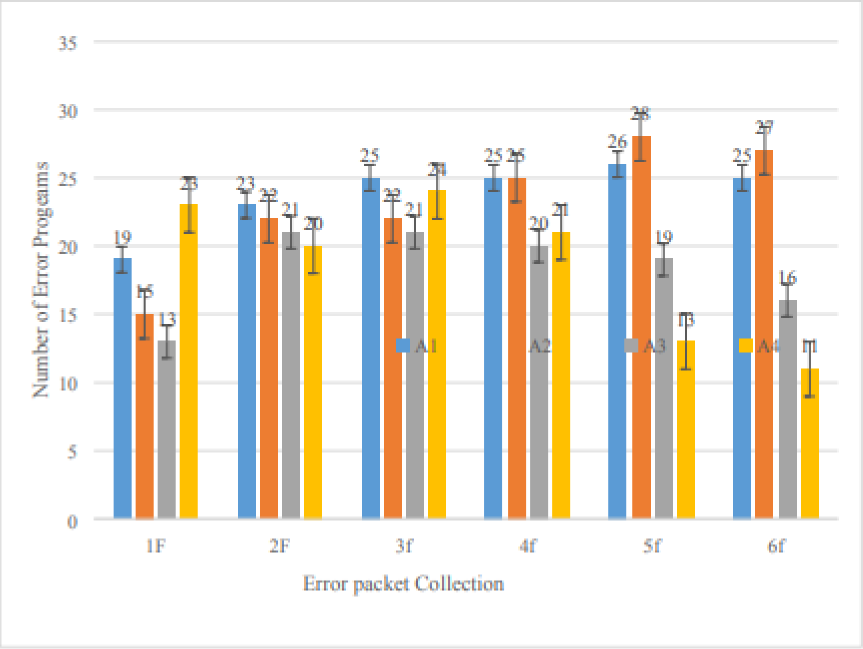

Figure. 1. Data map of experimental subjects

As shown in Figure 1, the selected four small C language tool programs contain hundreds of thousands or even tens of thousands of lines of code. Among these codes, there are more than a dozen to 30 error tests. Use cases are used to test different methods to make comparisons. The collection packages with errors from 1F to 6F each contain a different number of program errors. Table 2 shows that the 4 medium-scale C language tool programs selected in Table 1 are manually input to add errors, and the costs and normal costs of each version and each method are analyzed, using two Comparison of costs obtained by different methods.

Table. 2. Comparison of the cost of mislocation

Program	method(%)	1F	2F	3F	4F	5F	6F
A1	Sequential	3.75%	3.00%	1.70%	1.63%	1.43%	1.28%
A1	Increase	3.47%	0.64%	0.68%	0.95%	0.90%	0.88%
A2	Sequential	3.62%	3.16%	3.11%	3.04%	2.96%	2.78%
A2	Increase	3.50%	1.33%	0.85%	0.77%	0.67%	1.39%
A3	Sequential	1.08%	0.94%	0.91%	0.88%	0.80%	0.76%
A3	Increase	1.03%	0.33%	0.37%	0.23%	0.14%	0.11%
A4	Sequential	3.44	3.07	2.88	2.33	1.10	5.25
A4	Increase	3.15	2.15	2.79	2.53	3.98	5.94

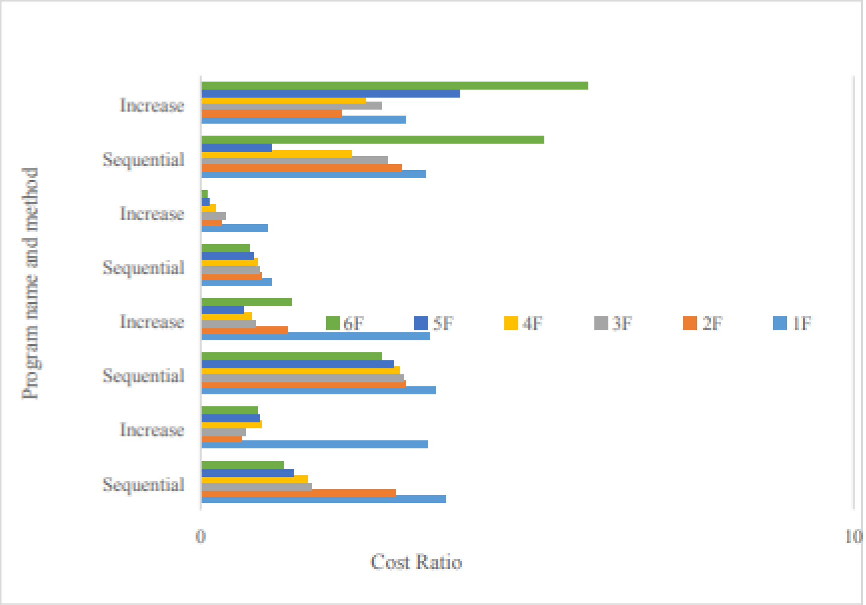

Figure. 2. Cost graph of each program and method

As shown in Figure 2, after manually injecting errors into these four medium-scale tool programs, the percentage of cost of errors in each version is reached. The first error located as the value range can be more intuitively expressed. There are two ways to locate the wrong price. For example, when locating the first test package F1, it contains 19 programs with errors.

5. Conclusion

The identification and location of software defects are essential to ensure the reliability and accuracy of software systems. However, identifying and locating these defects can be a challenging task, especially in complex software systems. To address this issue, this paper proposes the use of soft armor mining technology to locate software defects accurately. Soft armor mining technology is a type of data mining that uses algorithms to extract useful information from large data sets. The proposed method utilizes both static and dynamic software defect detection to locate software defects. Static software defect detection refers to the analysis of the source code to identify defects, while dynamic software defect detection refers to the analysis of the program's behavior during execution to identify defects. By using both methods, the proposed approach can identify defects in the software more accurately and comprehensively.

To obtain the location data set, the proposed method uses two different methods. The first method calculates the cost based on the first data set error. This method involves calculating the total number of errors and their locations in the software, which is used to estimate the cost of identifying and correcting the defects. The second method involves using a package in the A1 program that contains 23 error programs. The price paid for identifying and correcting these errors is an average of 3.00 for the first data set, which demonstrates the efficiency and cost-effectiveness of the proposed approach. The proposed soft armor mining technology-based approach has several advantages over traditional software defect detection methods. Firstly, it can locate defects more accurately and comprehensively, thereby reducing the risk of defects in software systems. Secondly, it can save time and resources by identifying defects early in the software development process. Finally, it can improve the overall quality of software systems by providing valuable insights into the areas of the software that require further refinement.

In conclusion, the proposed approach using soft armor mining technology to locate software defects is an effective and efficient method for identifying defects and errors in software systems. The use of both static and dynamic software defect detection, as well as two different methods to obtain the location data set, makes this approach a comprehensive and cost-effective solution for software defect location. As software systems become more complex, this approach is likely to become even more important in ensuring the reliability and accuracy of software systems.

References

[1] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert Syst. Appl.*, vol. 166, p. 114060, 2021.

[2] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods," *Eng. Appl. Artif. Intell.*, vol. 100, p. 104210, 2021.

[3] X. Xu, X. Yin, and X. Chen, "A large-group emergency risk decision method based on data mining of public attribute preferences," *Knowledge-Based Syst.*, vol. 163, pp. 495–509, 2019.

[4] S. N. Mohanty, E. L. Lydia, M. Elhoseny, M. M. G. Al Otaibi, and K. Shankar, "Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks," *Phys. Commun.*, vol. 40, p. 101097, 2020.

[5] Z. S. Ageed *et al.*, "Comprehensive survey of big data mining approaches in cloud systems," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 29–38, 2021.

[6] J. T. de Souza, A. C. de Francisco, C. M. Piekarski, and G. F. do Prado, "Data mining and machine learning to promote smart cities: A systematic review from 2000 to 2018," *Sustainability*, vol. 11, no. 4, p. 1077, 2019.

[7] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data," *Knowledge-Based Syst.*, vol. 187, p. 104837, 2020.

[8] B. P. L. Lau *et al.*, "A survey of data fusion in smart city applications," *Inf. Fusion*, vol. 52, pp. 357–374, 2019.

[9] A. Javeed, S. Zhou, L. Yongjian, I. Qasim, A. Noor, and R. Nour, "An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection," *IEEE access*, vol. 7, pp. 180235–180243, 2019.

[10] G. Chen, P. Wang, B. Feng, Y. Li, and D. Liu, "The framework design of smart factory in discrete manufacturing industry based on cyber-physical system," *Int. J. Comput. Integr. Manuf.*, vol. 33, no. 1, pp. 79–101, 2020.

[11] Y. Pan and L. Zhang, "A BIM-data mining integrated digital twin framework for advanced project management," *Autom. Constr.*, vol. 124, p. 103564, 2021.

[12] A. Livera, M. Theristis, G. Makrides, and G. E. Georghiou, "Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems," *Renew. energy*, vol. 133, pp. 126–143, 2019.

[13] E. Moghadas, J. Rezazadeh, and R. Farahbakhsh, "An IoT patient monitoring based on fog computing and data mining: Cardiac arrhythmia usecase," *Internet of Things*, vol. 11, p. 100251, 2020.

[14] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm Evol. Comput.*, vol. 54, p. 100663, 2020.

[15] H. A. Issad, R. Aoudjit, and J. J. P. C. Rodrigues, "A comprehensive review of Data Mining techniques in smart agriculture," *Eng. Agric. Environ. Food*, vol. 12, no. 4, pp. 511–525, 2019.

[16] W. Chen *et al.*, "Groundwater spring potential mapping using artificial intelligence approach based on kernel logistic regression, random forest, and alternating decision tree models," *Appl. Sci.*, vol. 10, no. 2, p. 425, 2020.

[17] N. O. Alsrehin, A. F. Klaib, and A. Magableh, “Intelligent transportation and control systems using data mining and machine learning techniques: A comprehensive study,” *IEEE Access*, vol. 7, pp. 49830–49857, 2019.

[18] M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, “Data mining and machine learning methods for sustainable smart cities traffic classification: A survey,” *Sustain. Cities Soc.*, vol. 60, p. 102177, 2020.

[19] A. S. Albahri *et al.*, “Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (COVID-19): a systematic review,” *J. Med. Syst.*, vol. 44, pp. 1–11, 2020.

[20] S. Qi *et al.*, “Review of multi-view 3D object recognition methods based on deep learning,” *Displays*, vol. 69, p. 102053, 2021.

[21] S. Shakya, “A self monitoring and analyzing system for solar power station using IoT and data mining algorithms,” *J. Soft Comput. Paradig.*, vol. 3, no. 2, pp. 96–109, 2021.

[22] Y. Zhao, C. Zhang, Y. Zhang, Z. Wang, and J. Li, “A review of data mining technologies in building energy systems: Load prediction, pattern identification, fault detection and diagnosis,” *Energy Built Environ.*, vol. 1, no. 2, pp. 149–164, 2020.

[23] G. Li *et al.*, “Research on the natural language recognition method based on cluster analysis using neural network,” *Math. Probl. Eng.*, vol. 2021, pp. 1–13, 2021.

[24] P. Duan, Z. He, Y. He, F. Liu, A. Zhang, and D. Zhou, “Root cause analysis approach based on reverse cascading decomposition in QFD and fuzzy weight ARM for quality accidents,” *Comput. Ind. Eng.*, vol. 147, p. 106643, 2020.

[25] S. Wu, J. Liu, and L. Liu, “Modeling method of internet public information data mining based on probabilistic topic model,” *J. Supercomput.*, vol. 75, pp. 5882–5897, 2019.