Research Article

Hong Wang and Limin Yuan*

Software engineering defect detection and classification system based on artificial intelligence

https://doi.org/10.1515/nleng-2022-0042
received December 08, 2021; accepted June 10, 2022

Abstract: With the increasing reliance on automatic software-based applications, it is important to automate the classification of software defects and ensure software reliability. An automatic software defect classification system based on an expert system is proposed in this article. In this method, DACS first determines the category of software defects through the selection of typical features, then reduces the spatial knowledge base searched by the inference engine and selects the characteristics of a certain type of defect. Make a selection, determine the name of the defect, and finally select different causes and prevention methods for the defect as needed. The DACS structure was built, and the experiment showed that the AI system took 15 s to complete, whereas the traditional mechanism took 48 s; the accuracy of the AI was 99%, whereas the accuracy of the traditional mechanism was only 68%. According to the aforementioned experimental results, the recognition accuracy of the proposed research scheme is higher than that of the traditional mechanism. Hence, the time required to solve the problem of software engineering defect detection and classification is less than that of the traditional mechanism.

Keywords: expert system, software defect detection, software defect classification

1 Introduction

Humans have been committed to making computer technology increasingly intelligent. It is a comprehensive discipline involving computers, control, linguistics, neurology, psychology, and philosophy. At the same time, artificial intelligence (AI) is also a subject with strong vitality. It tries to change human thinking and living habits, extend and liberate human intelligence, and will also lead mankind to a new era of scientific and technological development [1]. The development of AI also faces multiple problems, such as the core problem of AI: what is the difference between AI and “human nature”? What is the direction of AI? The conclusions drawn from a particular topic or only one aspect of AI research are worth discussing [2]. Only by using the dialectical materialist worldview and methodology, as well as computer expertise, can we break through the bottleneck and identify and solve various problems in the development of artificial intelligence technology [3]. After one-sided emotional negation, artificial intelligence ethics research has become increasingly scientific and mature in recent years. At present, with the rapid development of computer science and technology, as well as the ubiquitous use of software, the probability of errors or defects in the software development process is increasing. Any software model has a variety of software defects, and the automatic classification of software defects is particularly important to ensure their reliability. It is likely that the use of AI products will cause fear, but it is our human responsibility not to let that scene happen. We set a unique identification code for each intelligent robot to track its every move in real time. If the robot has any action or other behavior that the programmer does not set in any link, we should control it the first time to prevent it from causing harm to humans. At the same time, it can also prevent people with ulterior motives from copying artificial intelligence, just like the Shanzhai phones we have experienced. A mobile
phone full of rogue software is designed to steal a user’s private information and cause personal and property losses. As AI robots become more integrated into human life, more experts commit to the ethics of AI. Especially since 2005, some research papers and monographs on “robot ethics” have shown a steep rise in trend year by year [4]. The research on artificial intelligence ethics mainly focuses on AI safety, moral status, and environmental ethics. The mature field at this time is the harm brought by artificial intelligence to the environment because environmental problems are already apparent in our daily lives [5].

1.1 Literature review

On the premise of respecting the development of human scientific and technological civilization at the present stage, the thinking scope of intelligence philosophy is further reduced to ethical thinking; the development direction and prospects of artificial intelligence are judged with a more systematic ethical basis; and the ethical basis is used to give solutions [6]. The emergence of artificial intelligence technology is to imitate human intelligence, so ethical issues should be discussed as the most important problem in the development of artificial intelligence technology. It is advocated that, under any circumstances, scientific development research should be carried out on the premise of scientific development [7]. I believe that this is also the most needed in the process of AI development, as well as integrating AI technology and ethical issues as much as possible [8]. There have been many works and conclusions on the Marxist concept of science and technology, ethics, and philosophy of artificial intelligence. Through the worship of literature, I found that due to my limited academic ability, I could not fully understand the content of the literature, and there was no way to integrate all the views of the academic circle. Therefore, the research on the problems in the article is not comprehensive enough and lacks discussion depth too. At present, scientific research into artificial intelligence technology is still in its initial stages, and the specific form of artificial intelligence has not been finalized. Therefore, the final research on how to deal with ethical dilemmas is basically at a relatively shallow institutional level [9]. The expert system (expert system) is a branch of artificial intelligence. The expert system refers to the process of reasoning and judging data and problems entered by users based on knowledge in a knowledge database. The knowledge base is provided by experts, which contains specialized knowledge and experience, and for complex problems, expert systems rely on the theory of knowledge acquisition and knowledge representation. The expert system is designed to simulate the decision process of human experts [10].

1.2 Article contribution

Based on the current research, this article proposes AI-based software engineering defect detection and classification system research. The article first introduces the software defect classification, then builds the structure of DACS and creates a software defect database with SQLServer that includes two tables: the control and data flow-type defect table; the structured and non-functional defect table. This article proposed that the software defect classification can be done by using the problem information of the software tester during the test to analyze the wrong classification information. The conclusion is observed based on existing and known data and knowledge, gradually through positive logic. Experimental results show that the proposed AI system outperforms the traditional mechanism’s accuracy. The proposed DACS method first determines the category to which software defects belong by selecting typical characteristics and
then reduces the space of the reasoning machine search knowledge base. The proposed method then selects the name of the defect from among the characteristics of a certain kind of defect. It finally selects the different causes and prevention methods for a certain defect according to its needs [14]. It is shown in Figure 1.

1.3 Article organization

The rest of this article is structured as follows: an overview of the software defect classification is presented in Section 2, followed by results and analysis in Section 3. Section 4 presents the article’s conclusion.

2 Method

2.1 Overview of the software defect classification

With the rapid development of software engineering, especially in the process of software development, defects are not only limited to programs and code but also have a more or less impact on software quality in the software development process and the management structure of the organization. Researchers began to focus on the impact of software processes on software defects. There is a need to classify software flaws in order to analyze the causes of software defects and then improve the software process. This also prevents software defects from improving the quality of software and the software capability maturity of the organization [15,16]. Software classification should meet the following requirements: accurate classification of software defects; no overlap between the classification of software defects; and a classification system that includes all existing defect types; classification should be organically combined with the life cycle of software and classifying software defects from the perspective of software process [17,18]. The complete process of artificial intelligence-based classification is depicted in Figure 2.

2.2 Structure of the DACS

The basic idea of human experts in conducting software defect classification is to use the problem information of...
the software tester during the test to analyze the wrong classification information [19,20]. DACS simulates the basic idea of human problem-solving experts and consists of several modules of the user interface, knowledge base, reasoning machine, and interpreter. Its structure is shown in Figure 3.

3 Results and analysis

3.1 Knowledge base

The knowledge base is used to store expert-level domain knowledge, which contains a large number of facts and rules in a certain field, and all problems also rely on the knowledge base. There are two main types of knowledge in the knowledge base: one is factual knowledge, which is common knowledge; the other is enlightening knowledge, which is the summary of the personal experience of experts in the field. The knowledge in DACS’s knowledge base is derived from the orthogonal defect classification developed by IBM (Orthogonal Defects Classification [ODC]). The SQL server database system makes it easy to store, manage, and query data and to establish a software defect database. This database includes two tables: the control and data flow-type defect table; and the structured and non-functional defect table.

The control and data flow types include the following four types of defects:

- Inspection class refers to the error caused by missing or incorrect parameters or in the verification condition statement of the data.
- Algorithm and method class refers to the efficiency or correctness of the algorithm that affects the software development.
- Timing class refers to the error or wrong resource serialization technology.

Structured and non-functional classes include functional and object classes; I/O interface classes; and associated classes. The fields in each table have the same data type as well, for query convenience. For example, the table structure of the control and data flow class defect table is shown in Table 1.

The tree structure is shown in Table 2. This tree structure ensures the system can determine large branches during inference, such as whether the defect belongs to control and data flow or structured and non-functional categories. Then determine small branches, such as initialization and assignment, tests, algorithms, and methods. Therefore, after dividing the knowledge

<table>
<thead>
<tr>
<th>Field</th>
<th>Data type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>C</td>
<td>6</td>
</tr>
<tr>
<td>Type</td>
<td>C</td>
<td>10</td>
</tr>
<tr>
<td>Characteristic</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>Source</td>
<td>M</td>
<td>4</td>
</tr>
<tr>
<td>Preventive</td>
<td>M</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Table structure of control and data flow class defects
organization structure into multiple branches, there is no need to generate rules matching each branch, which greatly reduces the difficulty of searching the knowledge base [21,22].

Figure 4 depicts the different phases of software development along with the percentage of defects involved in each phase. The expected percentage to overcome these defects is also depicted in Figure 4.

### 3.2 Knowledge reasoning machine

The reasoning machine is driven by internal procedures and is the institution of the expert system-solving process. The inference strategy is mainly associated with the forward reasoning strategy, the reverse reasoning strategy, the mixed reasoning strategy, the conflict resolution strategy, the two-way reasoning strategy, and the meta-control strategy. The system uses a positive reasoning method. Positive reasoning has also become data-driven. Its basic idea is to understand the existing and known data and knowledge by gradually reasoning through positive logic. DACS first determines the category of software defects they belong to through the selection of typical characteristics, which reduces the space of reasoning machine search knowledge base. Then, he selects among the characteristics of a certain kind of defect, determines the name of the defect, and finally selects the different causes and prevention methods of a certain defect according to his needs. For each user’s selection of the defect feature, there will be an interpreter exporting the category or name information of the defect to the user.

### 3.3 User interface and interpreter

The user interface may be used to receive information from the user or send information to the user, and the user interface can serve as both a user input interface and a user output interface to transmit the information to the inference machine, which is another key component of the expert system. When the reasoner returns the useful information drawn from the knowledge base, the user interface retransmits the resulting information to the user in the appropriate form. The user interface provides communication for users and systems, including using keyboards and screens as input and output devices;

<table>
<thead>
<tr>
<th>Algorithms and methods</th>
<th>Initialization and assignment classes</th>
<th>Inspection class</th>
<th>Time series class</th>
<th>Functions and object classes</th>
<th>I/O interface class</th>
<th>Affiliate class</th>
</tr>
</thead>
<tbody>
<tr>
<td>The control and data flow-type defect</td>
<td>Structured and nonfunctional defects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4:** Different phases of software development.
questions and answers that support users and expert systems; identifying cognitive differences between users and expert systems; and providing a flexible and friendly interface for users. The responsibility of the interpreter is to explain the operation process and results of the system to the user and explain the conclusions given by the reasoning machine, its performance increases the trust and transparency of the expert system, improves the user acceptance of the conclusions obtained, and thus improves the user experience. The function of the knowledge acquisition machine converts the common sense knowledge in a certain field and the experiential knowledge of the field expert into a form that the computer can be recognized and utilized to edit the knowledge in the knowledge base. In different expert systems, there is no difference in knowledge acquisition functions and implementation methods. Some systems edit knowledge into the knowledge base through knowledge editing software, and some systems have learning functions. In the process of system operation, self-summary obtains new knowledge. The DACS system uses the experience of software testers and orthogonal defect classification by IBM as an important source of the knowledge base. This article uses the traditional manual inspection and the same software to record the time and results, respectively. The results are shown in Figure 5.

From the experimentation, it is revealed that DACS first determines the category to which software defects belong through the selection of typical characteristics, reducing the space of the reasoning machine search knowledge base. The selection among the characteristics of a certain kind of defect is done along with the determination of the name of the defect, thereby selecting the different causes and prevention methods of a certain defect according to the needs. The experimental results show that the AI system takes 15 s with 99% AI accuracy, whereas the traditional mechanism takes 48 s with only 68% accuracy.

4 Conclusions

This article proposes research on software engineering defect detection and classification system. It first introduces the software defect classification, creates the structure of DACS, and establishes a software defect database with SQLServer, which includes two tables: the control and data flow-type defect table; the structured and non-functional defect table. The basic idea of human experts in conducting software defect classification is to use the problem information provided by the software tester during the test to analyze the wrong classification information. The conclusion is based on existing and known data and knowledge, gradually through positive logic. Experimental results show that the AI system takes 15 s with 99% AI accuracy, whereas the traditional mechanism takes 48 s with only 68% accuracy. DACS first determines the category to which the software defects belong through the selection of typical characteristics, reduces the space of reasoning machine search knowledge base, then selects among the characteristics of a certain kind of defect, determines the name of the defect, and thereby selects the different causes and prevention methods of a certain defect according to the needs. The future scope of this article is to study deeply the software engineering
defect detection and classification system under other artificial intelligence.

**Acknowledgment:** This research was supported by S&T Program of Hebei, Research on the innovation of fault tolerance and error correction mechanism in scientific research activities in Colleges and Universities, Project No: 215576157D.

**Author contributions:** All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

**Conflict of interest:** The authors state no conflict of interest.

**References**


