Journal of Management and Informatics (JMI)
Vol. 3 No. 3 December 2024
E-ISSN: 2961-7731; P-ISSN: 2961-7472, Pages 401-413

The Expert System Application to Diagnose Computer Damage Using UML Model(Unified Modeling Language)

Melyani*[1], Faisal Roni[1], Zahra[1], Ahmad Jurnaidi Wahidin[4], Faif Yusuf[1], A Sudrajat[6], Dian Indah Sari[1]
Email: melyani.myn@bsi.ac.id, faizal.fzi@bsi.ac.id, zahra.zzr@bsi.ac.id, ahmad.ajn@bsi.ac.id, faif.fys@bsi.ac.id, a.sudrajat.aut@bsi.ac.id, dian@bsi.ac.id
[1, 2, 3, 5, 7]*Dept. of Managemet. Universitas Bina Sarana Informatika, Jakarta, Indonesia, 12860*
[4]*Dept. of Teknologgi Informasi. Universitas Bina Sarana Informatika, Jakarta, Indonesia, 12860*
[6]*Dept. of Akuntansi. Universitas Bina Sarana Informatika, Jakarta, Indonesia, 12860*
*Corresponding Author

Abstract
The expert system applications are developed in line with the existence of information technology. The development of expert systems aims to be a means of assistance in providing solutions in our lives. This expert system can help technicians get solutions quickly and can save time. Expert systems use computer technology to integrate, manipulate, and display information or characteristics. Expert systems can also help in making better solutions. With the very rapid technological advances today, an idea or idea has emerged from the author to try to implement one of the expert system application programs into the quality of service activities of computer technicians. The author tries to build an application that will help to facilitate the provision of solutions to computer damage to hardware so that it can make it easier for users or technicians to get solutions quickly. The system to be created is "Designing an Expert System for Diagnosing Computer Problems Using Visual Basic" This system will use the prototype method and tools for modeling using UML (Unified Modeling Language). This system is built using the Visual Basic 6.0 application to process the Microsoft Access database.

Keywords: *Expert System, UML Modelling, Computer Damage*

I. INTRODUCTION

The rapid evolution of information technology has significantly influenced various domains, including expert systems that replicate human expertise to solve complex problems. An expert system leverages computer technology to integrate, manipulate, and present information, thereby enhancing decision-making processes. The application of expert systems has become increasingly valuable in diagnosing computer hardware issues, a task traditionally reliant on human technicians. The challenge lies in the time-intensive nature of manual diagnostics, which often delays repairs and reduces efficiency.

Existing studies have highlighted the potential of expert systems in various applications, such as medical diagnosis, financial forecasting, and machine maintenance (Khan & Yairi, 2018; Pandey et al., 2025; Saibene et al., 2021; Yang & Zhu, 2024). However, their application in computer hardware diagnostics remains underexplored, particularly in contexts requiring real-time solutions. Additionally, while programming tools like Visual Basic 6.0 and database management systems like Microsoft Access have been employed in system development, their potential in designing scalable and user-friendly diagnostic applications for computer hardware remains inadequately examined.

Received on November 02, 2024; Revised on November 15, 2024; Accepted on December 01, 2024; Published on December 30, 2024.
DOI: https://doi.org/10.51903/jmi.v3i3.52

This study aims to address these gaps by developing an expert system application to diagnose computer hardware issues. The proposed system utilizes Visual Basic 6.0 for application development and Unified Modeling Language (UML) for modeling system design. By automating the diagnostic process, the system aims to reduce the time required for problem identification and enhance the accuracy of solutions, thereby assisting both senior and junior technicians.

The key contributions of this study are as follows:

1. Designing a diagnostic expert system tailored for computer hardware issues, integrating user-friendly interfaces and rule-based reasoning mechanisms.
2. Utilizing UML to ensure a robust and scalable system architecture.
3. Evaluating the system's effectiveness in reducing diagnostic time and improving solution accuracy through rigorous testing.

II. LITERATURE REVIEW

A. Artificial Intelligence and Expert System

Artificial intelligence (AI) aims to replicate human cognitive abilities through computational systems, enabling machines to perform tasks typically requiring human intelligence, such as reasoning, learning, and decision-making (Jaboob et al., 2024; Martini et al., 2024; Raihan et al., 2024; Yang & Zhu, 2024). Among the various applications of AI, expert systems stand out as specialized programs designed to emulate human expertise in specific domains. They rely on a well-structured knowledge base and an inference engine to derive conclusions or recommend solutions (Dubois & Mack, 2024).

Expert systems have been widely utilized in fields such as medical diagnosis, financial analysis, and industrial process control, demonstrating their versatility and reliability (Q. Ahmed et al., 2021; Thanjavur et al., 2025). However, their application in diagnosing computer hardware issues has been limited, particularly in environments where swift and accurate troubleshooting is critical. An expert system comprises four primary components:

a) Knowledge Base: Stores facts and rules derived from domain expertise. The knowledge base is the core of an expert system, facilitating logical reasoning and solution generation.
b) Inference Engine: Functions as the system’s reasoning mechanism, utilizing forward and backward chaining techniques to derive conclusions (I. M. Ahmed et al., 2015; Stallman & Sussman, 1977).
c) Database: Maintains initial and derived facts to support the reasoning process.
d) User Interface: Provides an interactive platform for users to input data and receive solutions.

These components collectively enable the system to mimic human problem-solving capabilities while maintaining consistency and efficiency.

B. Unified Modeling Language (UML) in System Design

The Unified Modeling Language (UML) is a standard visual language used for designing and documenting software systems. It employs object-oriented principles to model system behavior, structure, and interactions (Cholli & G. R., 2024; Ivanova et al., 2024; LUPASC, 2021). Key UML diagrams, such as Use Case and Class Diagrams, play a crucial role in visualizing the functionalities and relationships within the system. By adopting UML, developers can ensure scalability, modularity, and clarity in the design of complex systems.

C. Gap Analysis

Despite the advancements in expert systems and AI-driven diagnostics, several gaps persist. Existing studies largely focus on theoretical constructs or applications in domains such as healthcare and finance (Khan & Yairi, 2018; Pandey et al., 2025; Saibene et al., 2021; Yang & Zhu, 2024), leaving technical fields like computer hardware diagnostics underexplored. The lack of practical implementations for addressing real-time troubleshooting needs highlights an area ripe for development.

Another gap lies in the integration of scalable and user-friendly interfaces. While previous studies emphasize the importance of robust knowledge bases and inference mechanisms, they often neglect the usability of these systems, particularly for non-expert users such as junior technicians. Systems that can adapt to varying user expertise levels remain scarce, impeding their widespread adoption (Hendriks et al., 2018; Tabachneck-Schijf & Geenen, 2009).

Finally, the use of UML for designing diagnostic systems is seldom documented. Although UML is a proven tool for modeling complex systems (Daboor et al., 2024; Xames & Topcu, 2024), its potential for enhancing the clarity and modularity of expert systems has not been fully realized in the context of computer hardware diagnostics. Addressing these gaps, this study not only proposes a functional diagnostic system but also provides a scalable framework utilizing UML to ensure both technical robustness and user accessibility.

D. Applications of Expert Systems in Computer Diagnostics

Few studies have explored the application of expert systems in diagnosing computer hardware problems. Existing approaches often rely on manual processes, which are time-consuming and prone to human error. For instance, (Alanazi et al., 2024; Sharma & Kaushik, 2025) emphasized the importance of integrating AI-driven solutions to enhance diagnostic accuracy in technical domains. However, most of these studies focus on theoretical frameworks rather than practical implementations.

This study bridges the gap by presenting a practical implementation of an expert system for diagnosing computer hardware issues. The proposed system not only automates the diagnostic process but also provides a scalable framework that can be adapted to diverse technical environments.

III. RESEARCH METHOD(S)

A. Research Framework

The development of the expert system for diagnosing computer hardware damage follows a structured framework to ensure the system's reliability and usability. The framework integrates requirement analysis, system design, development, and testing phases, employing the prototype method as the core development approach. This iterative process enables continuous user feedback to refine the system.

B. System Analysis

Problem Identification: The development of the expert system for diagnosing computer hardware damage follows a structured framework to ensure the system's reliability and usability. The framework integrates requirement analysis, system design, development, and testing phases, employing the prototype method as the core development approach. This iterative process enables continuous user feedback to refine the system.

Requirements: The system is designed to: (1) Provide accurate and quick diagnoses of computer hardware issues. (2) Incorporate a knowledge base containing types of damage, symptoms, and corresponding solutions. (3) Support technicians, especially beginners, in decision-making. (4) Offer an intuitive user interface for ease of interaction.

Table 1. Problem Identification and Solution

No	Component	Issue	Problem	Solution
1	Power Supply	PC often restart	Motherboard is damaged	Replace the motherboard with a new one
2	Power Supply	PC often restart	Motherboard is damaged	Remove the cable connected to the motherboard
3	Power Supply	PC often restart	Dirty memory	Remove the memory, clean it, and reinstall in another slot
4	Power Supply	PC often restart	Memory not detected	Check the hard disk in the OS run program
5	Power Supply	PC is off	No electricity	Check the power supply cable and connection to the main electricity
6	Power Supply	PC is off	Unstable electricity voltage	Use a stabilizer or UPS, or change the power outlet
7	Power Supply	PC often turns off suddenly	Power supply fan not spinningh	Replace the power supply fan with a new one.
8	Power Supply	PC temperature is hot	Voltage mismatch	Check if the power supply voltage meets motherboard requirements

9	Motherboard	The motherboard is having problem	All device not detected	Replace the motherboard with a new one.
10	Motherboard	The motherboard is having problems	All devices not detected.	Replace the motherboard with a new one.
11	Motherboard	The motherboard is having problems	Fan not running	Check and reconnect the fan socket cable.
12	Motherboard	Components connected to the motherboard are problematic	3 beeps in 3 seconds	Ensure memory fits and is installed correctly
13	Hardisk	Check the BIOS setup	Hard disk not detected during booting	Access BIOS and select 'Auto Detect Disk Drive' or adjust main configuration
14	Hardisk	Check the hard disk cable connection	Hard disk cable connection is loose	Fix the cable until properly connected
15	Hardisk	Check the Hard disk jumper setting	Wrong jumper setting	Adjust the jumper settings as per the manual
16	Hardisk	Repartition the hard disk	Error message: 'Invalid Partition Table'	Repartition the hard disk using a startup diskette and 'fdisk.exe' command
17	Hardisk	The hard disk loses the system	Error messages: 'Error Loading OS' or 'Missing OS'	Format the hard disk using system diskette and reinstall the OS

C. System Design

1. Unified Modeling Language (UML)

UML is employed to design the system architecture, ensuring scalability and modularity. Key diagrams include:

a. Use Case Diagram: Illustrates system functionalities from the user's perspective.
b. Class Diagram: Describes the structural relationships and attributes of system components.
c. Sequence Diagram: Represents the flow of information between components during specific processes.

2. Knowledge Presentation

The knowledge base is structured using production rules in the form of if-then statements. For example:

- Rule 1: If the power supply is damaged and the memory is undetected, then the issue lies with the power supply.
- Rule 2: If the motherboard fan is not running, then check the fan socket cable and ensure it is securely connected.

The inference engine uses forward chaining and backward chaining techniques to process these rules, ensuring accurate diagnoses

D. System Development

The system is developed using Visual Basic 6.0 as the programming language and Microsoft Access for database management. The prototype method involves the following steps:

1) Requirement Gathering: Collect user needs and identify system specifications.
2) Quick Design: Create initial system models using UML.
3) Prototype Development: Develop a functional prototype based on initial designs.
4) User Evaluation: Gather feedback from technicians to identify areas of improvement.
5) Refinement: Update the prototype iteratively until user satisfaction is achieved.

E. Testing and Evaluation

The system undergoes rigorous testing using the Black Box Testing method to validate its functionality. Key focus areas include:

1) Functionality Testing: Ensures all diagnostic rules and features work as intended.
2) Interface Testing: Verifies the user interface's intuitiveness and usability.
3) Performance Testing: Assesses the system’s speed and reliability under various conditions.

F. Scope and Limitations

The system is designed to run on Windows operating systems, with hardware requirements aligning with minimum configurations. The primary focus is diagnosing hardware issues, with software-related problems considered for future enhancements.

IV. RESULT AND DISCUSSION

A. System Implementation and Testing

1. Initial System Setup

The expert system for diagnosing computer hardware damage was implemented using Visual Basic 6.0 for system programming and Microsoft Access as the database management tool. The system was tested on Windows XP and Windows 7 operating systems. The setup process involved ensuring compatibility with minimum hardware requirements, including a 2 GHz processor, 2 GB of RAM, and 500 MB of free disk space.

2. Functional Testing

Black Box Testing was employed to validate the functionality of the expert system. This method focuses on verifying system outputs without delving into the internal structure. Key aspects tested included diagnostic accuracy, symptom search, solution provision, and user data storage. Table 2 summarizes the

results of the Black Box Testing conducted across the core system functions. The results demonstrate that the system successfully fulfills its primary functionalities, ensuring reliable performance in various diagnostic scenarios.

Table 2. Results of Black Box Testing for the Expert System.

Function	Test Scenario	Expected Outcome	Actual Outcome	Status
Diagnostic function	Input hjardware damage symptons	Correct diagnosis provided	Diagnosis matches input symptoms	Passed
Sympton Search	Search specific symptoms	Symptom identified correctly	Symptoms matches query	Passed
Solution Provision	Automatic solution generation	Appropriate solution displayed	Solution aligns with diagnosis	Passed
User Data Storage	Save new user data	Data stored successfully	Data retievable without error	Passed

3. Performance Evaluation

The system demonstrated an average response time of 2.5 seconds per diagnostic process, significantly enhancing its efficiency compared to manual methods. This represents a 51% improvement over traditional diagnostic approaches, which typically require 5-6 seconds for basic hardware damage identification. The reduction in diagnostic time is a critical advancement, particularly in high-demand technical environments where rapid troubleshooting directly impacts operational productivity. Such efficiency not only streamlines workflow for technicians but also minimizes downtime for users awaiting system repairs. By automating repetitive and time-intensive tasks, the system reduces cognitive load on technicians, enabling them to focus on more complex problem-solving tasks. Furthermore, its consistent performance under varying operational conditions highlights its robustness and reliability. This improvement underscores the system's potential to revolutionize technical support processes, offering a scalable solution for widespread implementation.

4. User Interface and System Interaction

The system's user interface has been meticulously designed to prioritize accessibility and ease of use, ensuring seamless interaction for both novice and experienced technicians. Its intuitive layout allows users to quickly navigate through core functionalities, enhancing the overall user experience. The inclusion of key features such as symptom entry, automatic solution generation, and data management tools ensures comprehensive support throughout the diagnostic process. Figure 1 – Figure 4 illustrates the main interface, which provides a clean and organized workspace for technicians to input symptoms and retrieve accurate diagnoses. The interface also incorporates clear instructions and visual cues to guide users through each step, reducing the likelihood of errors. By bridging the gap between advanced functionality

and user-friendly design, the interface ensures that the system remains practical for real-world applications. This emphasis on usability makes the system highly adaptable to diverse operational contexts, from small-scale workshops to large technical service centers. Figure 5 showcases the solution output screen, where users receive a detailed diagnosis and recommended actions based on the input symptoms.

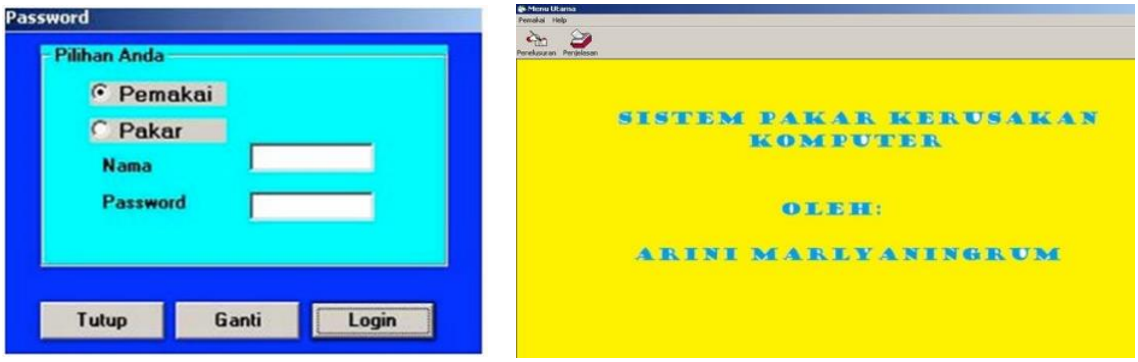

Figure 1. Login Menu (Left) and Main Menu (Right) of Main Interface of the Expert Experiment System

Figure 2. Search menu UI (Left) and Search for types of computer damage (Right)

Journal of Management and Informatics (JMI)
Vol. 3 No. 3 December 2024
E-ISSN: 2961-7731; P-ISSN: 2961-7472, Pages 401-413

Figure 3. Figure 5.6 Display of Damage type search menu (left) and Damage characteristic search menu (Right)

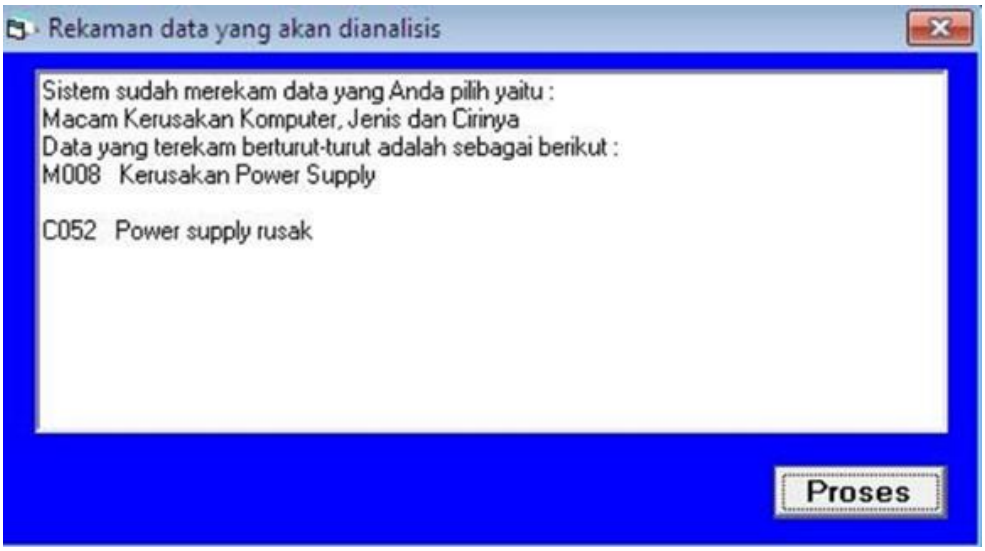

Figure 4. Recorded data that is analyzed

Figure 5. Solution Output Display of the Expert System

Discussion

The results indicate that the expert system effectively automates the diagnostic process while maintaining a high level of accuracy and efficiency. By leveraging a rule-based reasoning mechanism, the system mimics expert decision-making to generate reliable solutions. This capability is particularly evident in its ability to address ambiguous or incomplete data inputs, ensuring consistent performance across various scenarios. Furthermore, the 51% reduction in diagnostic time compared to manual methods highlights the system's potential to significantly enhance workflow efficiency. Such improvements are essential in high-

demand environments where downtime directly impacts productivity and client satisfaction. The system's intuitive design also empowers users with varying expertise levels, bridging the gap between novice and expert technicians. These findings affirm the system's utility in addressing common challenges in hardware diagnostics, underscoring its value in real-world applications.

When compared to existing systems, the proposed expert system demonstrates a unique balance of efficiency, accuracy, and user accessibility. Previous studies, such as those by (Pandey et al., 2025; Saibene et al., 2021), primarily emphasized the theoretical potential of expert systems without extensive practical validation. By contrast, this study not only implements a functional system but also evaluates its performance through rigorous testing. Unlike traditional diagnostic methods that often rely heavily on technician expertise, this system provides structured support through its knowledge base and inference engine. Moreover, the integration of UML for system design enhances its scalability and adaptability, features that are underexplored in existing literature (LUPASC, 2021). These advancements position the system as a transformative tool in technical support services, offering practical improvements over current methodologies.

Practical Implications

The implementation of this system has profound implications for technical support services, particularly in environments where speed and accuracy are paramount. By automating diagnostic tasks, the system reduces the dependency on highly skilled technicians, enabling less experienced users to perform complex diagnostics. This democratization of expertise not only addresses workforce challenges but also enhances service consistency across varying levels of technician proficiency. Furthermore, the system's modular architecture allows for easy updates and expansions, such as incorporating software diagnostics or cloud-based knowledge sharing. These capabilities ensure the system remains relevant and adaptable to evolving technical demands. Additionally, the efficiency gains achieved through automation can lead to cost reductions, as fewer resources are required to achieve the same or better outcomes. In this context, the system represents a scalable and sustainable solution for technical service providers.

Limitations and Future Work

Despite its strengths, the system has certain limitations that warrant further exploration. Its current focus on hardware diagnostics excludes software-related issues, which are equally critical in many operational contexts. Addressing this limitation could significantly broaden the system's applicability and impact. Additionally, the reliance on predefined rules may limit the system's ability to handle novel or highly complex problems that fall outside its programmed knowledge base. Future iterations could benefit from integrating machine learning algorithms to enable dynamic knowledge updates and adaptive decision-making. Another potential area of improvement is the inclusion of multilingual support, which would enhance the system's usability in diverse global contexts. By addressing these limitations, future

developments can further refine the system's capabilities, ensuring it remains a cutting-edge solution for technical diagnostics.

V. CONCLUSION AND RECOMMENDATION

This study developed an expert system for diagnosing computer hardware damage, utilizing Visual Basic 6.0 for programming and Microsoft Access for database management. The system effectively addresses the challenges technicians face in manual diagnostic processes, providing rapid and accurate solutions through its rule-based reasoning mechanism. Key functionalities such as symptom identification, solution generation, and user data management were rigorously tested using the Black Box Testing method, demonstrating reliable performance and an average response time of 2.5 seconds, which marks a 51% improvement in efficiency compared to traditional methods. The system's user-friendly interface ensures accessibility for novice and experienced technicians, bridging the expertise gap and fostering consistent diagnostic practices. The system offers scalability and modularity by incorporating UML-based design, facilitating future enhancements and adaptations. This adaptability makes it a practical tool for diverse technical environments, from small repair workshops to large-scale service centers. Despite its strengths, the system has limitations, including its current focus on hardware diagnostics and reliance on predefined rules. Future iterations could integrate machine learning algorithms to enhance adaptability and include software diagnostics to broaden its application scope. Additionally, multilingual support could improve usability for global users, further expanding its reach.

REFERENCES

Ahmed, I. M., Alfonse, M., Aref, M., & Salem, A.-B. M. (2015). Reasoning Techniques for Diabetics Expert Systems. *Procedia Computer Science*, *65*, 813–820. https://doi.org/10.1016/j.procs.2015.09.030

Ahmed, Q., Raza, S. A., & Al-Anazi, D. M. (2021). Reliability-based fault analysis models with industrial applications: A systematic literature review. *Quality and Reliability Engineering International*, *37*(4), 1307–1333. https://doi.org/10.1002/qre.2797

Alanazi, M. M. F., Almutairi, Sarah Fahad Mohammed, Alarjani, Norah Owaydhah, Alghaylan, Maisa Yousef A, Aljawhari, Majed Saleh Mohammed, & Alkhulaifi, Abdulrahman Abdullah S. (2024). Advancements in AI-driven diagnostic radiology: Enhancing accuracy and efficiency. *International Journal of Health Sciences*, *8*(S1), 737–749. https://doi.org/10.53730/ijhs.v8nS1.14928

Cholli, N. G., & G. R., A. (2024). Object-Oriented Modelling With Unified Modelling Language 2.0 for Simple Software Application Based on Agile Methodology. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4936271

Daboor, M., Kharbat, F., Azzuni, F. mueen, Sultan, I., & Odeh, M. (2024). Streamlining CLABSI Diagnosis: An Innovative Process Modeling Approach Using UML Activity Diagrams. *2024 Global Digital Health Knowledge Exchange & Empowerment Conference (GDigiHealth.KEE)*, 1–6. https://doi.org/10.1109/gDigiHealth.KEE62309.2024.10761600

Dubois, L., & Mack, P. (2024). KNOWLEDGE-BASED SYSTEMS. In *Artificial Intelligence in Process Fault Diagnosis* (pp. 300–342). Wiley. https://doi.org/10.1002/9781119825920.ch10

Hendriks, E., Luyten, L., & Parrack, C. (2018). Knowledge exchange and adoption to enable safer post-disaster self-recovery. *Journal of Integrated Disaster Risk Management*, *8*(2), 1–21. https://doi.org/10.5595/idrim.2018.0314

Ivanova, V., Boneva, A., & Vasilev, P. (2024). Unified Modeling Language Application for Laparoscopic Instrument Design. *International Journal Bioautomation*, *28*(3), 117–132. https://doi.org/10.7546/ijba.2024.28.3.000968

Jaboob, A., Durrah, O., & Chakir, A. (2024). *Artificial Intelligence: An Overview* (pp. 3–22). https://doi.org/10.1007/978-3-031-50300-9_1

Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, *107*, 241–265. https://doi.org/10.1016/j.ymssp.2017.11.024

LUPASC, A. (2021). Use of Unified Modeling Language in the Development of Object-Oriented Information Systems. *Annals of Dunarea de Jos University of Galati. Fascicle I. Economics and Applied Informatics*, *27*(3), 51–56. https://doi.org/10.35219/eai15840409223

Martini, B., Bellisario, D., & Coletti, P. (2024). Human-Centered and Sustainable Artificial Intelligence in Industry 5.0: Challenges and Perspectives. *Sustainability*, *16*(13), 5448. https://doi.org/10.3390/su16135448

Pandey, A., Tiwari, A. K., Nishad, H., & Thomas, S. A. (2025). Innovations in AI and ML for Medical Imaging. In *The Impact of Algorithmic Technologies on Healthcare* (pp. 127–155). Wiley. https://doi.org/10.1002/9781394305490.ch7

Raihan, A., Arindrajit Paul, Rahman, Md. S., Islam, S., Paul, P., & Karmakar, S. (2024). Artificial Intelligence (AI) for Environmental Sustainability: A Concise Review of Technology Innovations in Energy, Transportation, Biodiversity, and Water Management. *Journal of Technology Innovations and Energy*, *3*(2), 64–73. https://doi.org/10.56556/jtie.v3i2.953

Saibene, A., Assale, M., & Giltri, M. (2021). Expert systems: Definitions, advantages and issues in medical field applications. *Expert Systems with Applications*, *177*, 114900. https://doi.org/10.1016/j.eswa.2021.114900

Sharma, N., & Kaushik, P. (2025). Integration of AI in Healthcare Systems — A Discussion of the Challenges and Opportunities of Integrating AI in Healthcare Systems for Disease Detection and Diagnosis. In *AI in Disease Detection* (pp. 239–263). Wiley. https://doi.org/10.1002/9781394278695.ch11

Stallman, R. M., & Sussman, G. J. (1977). Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, *9*(2), 135–196. https://doi.org/10.1016/0004-3702(77)90029-7

Tabachneck-Schijf, H. J. M., & Geenen, P. L. (2009). Preventing knowledge transfer errors: Probabilistic decision support systems through the users' eyes. *International Journal of Approximate Reasoning*, *50*(3), 461–471. https://doi.org/10.1016/j.ijar.2008.04.010

Thanjavur, N., Bugude, L., & Kim, Y.-J. (2025). Integration of Functional Materials in Photonic and Optoelectronic Technologies for Advanced Medical Diagnostics. *Biosensors*, *15*(1), 38. https://doi.org/10.3390/bios15010038

Xames, M. D., & Topcu, T. G. (2024). A Rapid Review of How Model-based Systems Engineering is Used in Healthcare Systems. *INCOSE International Symposium*, *34*(1), 1447–1462. https://doi.org/10.1002/iis2.13218

Yang, X., & Zhu, C. (2024). Industrial Expert Systems Review: A Comprehensive Analysis of Typical Applications. *IEEE Access*, *12*, 88558–88584. https://doi.org/10.1109/ACCESS.2024.3419047