DOI: 10.5281/zenodo.15512204

AUTOMATION OF IOT LABORATORY TASKS USING AI

Peter Procházka[31]

Abstract

The Internet of Things (IoT) represents an interdisciplinary approach encompassing electronics, programming, and network communication, placing high demands on assessing students' practical skills. This article analyzes existing tools used in IoT laboratories and identifies their limitations. We propose a solution that leverages artificial intelligence (AI) to automate the detection of hardware errors and software evaluation. AI models trained on image and programming data will enable efficient monitoring of component connections and code assessment. The expected outcomes include increased assessment accuracy, immediate feedback, and reduced instructor workload. Automating IoT laboratories thus brings significant advancements in the field of education.

Keywords

IoT Education, Automation, Artificial Intelligence, Error Detection, Software Assessment

1 Introduction

Practical learning is essential in understanding the complex Internet of Things (IoT) concepts. Practical, hands-on education is fundamental for grasping the intricacies of IoT as it bridges theoretical understanding with tangible, real-world applications (Atzori et al., 2010). IoT combines hardware devices, software platforms, and network communication, requiring students to develop interdisciplinary skills. Traditional assessment of work in IoT laboratories is time-consuming and requires thorough instructor evaluation. Automating this process with artificial intelligence (AI) can streamline assessments, provide students with immediate feedback, and enhance the overall efficiency of education. AI-powered educational assessment tools improve accuracy and efficiency and generate personalized feedback, helping educators adapt their teaching strategies to individual needs" (Owan et al., 2023). Artificial intelligence tools have demonstrated their potential to reduce the workload of educators by automating repetitive tasks and delivering timely feedback to enhance student learning outcomes (Holmes et al., 2019).

Our work aims to analyze the possibilities of automating IoT laboratories using AI and propose a solution that integrates hardware connection detection, code evaluation, and personalized feedback. This approach can overcome existing obstacles in assessment and optimize the student learning process.

[31] University of Economics in Bratislava / faculty of Economic Informatics, department of Applied Informatics, Dolnozemská cesta 1/b, 852 35 Bratislava 5, Slovakia, e-mail peter.prochazka@euba.sk

2 Current State and Challenges

The practical applications of AI in laboratory evaluation are multifaceted and encompass various fields. Using machine learning in defect detection will improve manufacturing processes and reduce product defects, ultimately increasing company customer satisfaction and profitability. Machine learning will also be applicable in other areas, such as medical diagnostics, security, and cybersecurity. With machine learning, monitoring and controlling the condition and quality of goods during production will be possible. In the automotive and engineering industries, defect detection systems will be utilized to inspect joints, welds, rivets, screws, coatings, or anomalies that might occur (Baraniak, 2024). For example, PathAI has developed AI systems for pathology analysis that can accurately diagnose various types of cancer. Another example is the use of AI in genetic laboratories, where machine learning is employed to identify genetic mutations and predict hereditary diseases (Benjamin, 2024).

Similarly, automating IoT laboratories with artificial intelligence will represent a crucial step toward enhancing the efficiency of education and evaluation. For instance, systems combining machine learning and image analysis will be able to detect errors in hardware configurations, significantly reducing the time required to verify students' practical assignments. As mentioned previously, IoT education encompasses knowledge from electronics, programming, and network communication, each of which requires specific approaches to assessment. Developing interdisciplinary skills in IoT demands a blend of theoretical foundations and practical experiences tailored to address the multifaceted nature of IoT systems (Xu et al., 2014). Therefore, the challenges associated with automation will span a broad range of closely interconnected problems.

2.1 Software Tools for IoT Education

IoT laboratories rely on various software tools that enable students to simulate, program, and test IoT solutions. One of the most commonly used tools is **Cisco Packet Tracer**, which provides an environment for simulating network communication and configuring IoT devices. This tool offers realistic modeling of network topologies and devices, but its main limitation is the lack of automated evaluation for complex configurations or software solutions.

TinkerCAD is often utilized for circuit modeling. It provides an intuitive graphical interface for creating circuit prototypes and simulating their behavior. This tool is ideal for beginners as it helps them easily understand the basics of electronics and IoT. However, its capabilities fall short for advanced analyses or evaluations of real-world circuits, limiting its usability in higher education environments.

The dominant tool for programming IoT devices is the Arduino IDE, which allows students to create and debug code for Arduino, ESP32, or NodeMCU microcontrollers. The Arduino IDE supports coding in a simple, C/C++-like language and offers a vast library base for working with sensors and actuators. Nevertheless, it lacks an automated mechanism to evaluate the effectiveness and correctness of code, often extending the time instructors spend on manual assessment.

2.2 Hardware Tools and Their Limitations

IoT education inherently requires working with physical devices. Commonly used microcontrollers such as **Arduino**, **NodeMCU**, or **ESP32** provide students a platform for executing practical projects. These devices are equipped with GPIO pins to connect sensors and actuators, which students use to simulate real-world IoT solutions. However, configuring these devices correctly can be challenging for beginners.

One of the biggest challenges is detecting faulty connections. AI can identify and predict hardware failures by analyzing patterns in system data, which reduces downtime and enhances system reliability" (Mahamuni et al., 2024). Detecting faulty connections in IoT systems often requires advanced AI-based image processing techniques to ensure reliability and efficiency (Minchev & Dimitrov, 2020). Improperly connected sensors can render the entire system non-functional. Students often overlook seemingly simple issues, such as component polarity (LEDs, capacitors, motors, etc.), or fail to identify the correct connection of GPIO pins. Manually assessing these configurations requires intensive instructor supervision, which is time-consuming and not scalable for larger groups. AI-enhanced laboratory systems can autonomously detect incorrect sensor setups and guide students through corrective actions, reducing dependency on instructors (Essa et al., 2023).

2.3 Deficiencies in Code Evaluation

Code evaluation in IoT laboratories will involve multiple aspects: syntax, logical correctness, efficiency, and documentation. Current code analysis tools, such as **Pylint**, **SonarQube**, and **PlatformIO**, will be able to identify syntax errors and fundamental formatting issues. However, these tools are limited in analyzing logical errors or optimizing IoT algorithms. For example, evaluating an energy management algorithm for battery-powered devices will require a deeper understanding of the context, which standard tools cannot automatically provide. AI algorithms integrated with IoT platforms optimize energy usage by predicting device performance and adjusting operations accordingly (Singh & Tomar, 2021). Advanced AI-driven tools use predictive analysis to identify more profound logical errors in code, helping developers to refine and optimize their programs (Nama, 2024). AI tools enhance code analysis by automating syntax checking and logical evaluation, enabling developers to identify inconsistencies and improve code quality efficiently (Nikolaidis et al., 2024).

Furthermore, there is no mechanism to provide students with personalized feedback on their work. AI could offer such feedback by analyzing individual error patterns in code and suggesting specific steps for correction. For syntax errors, it could provide examples of correct usage, while for logical errors, it could generate a visualization of the program flow and highlight the points of failure. AI could propose connection adjustments or identify missing components based on image analysis for hardware issues. IoT education involves an iterative process of improving solutions, where students need to understand their mistakes and learn how to fix them. Manual evaluation of this process is time-consuming and prevents immediate correction.

2.4 Lack of Automation Tools

Despite advancements in IoT education, existing tools fail to address automated evaluation needs comprehensively. Most available solutions are focused either on simulation or specific components of the evaluation process, lacking an integrated approach that combines hardware and software verification. Moreover, current systems often rely on manual intervention, which slows down the learning process.

These shortcomings result not only in increased demands on instructors but also in lower-quality feedback provided to students. Automating the evaluation of IoT projects using AI represents a promising approach to overcoming these challenges and enhancing the overall learning process.

3 Proposal for an AI-Powered Automation Solution

Automating the evaluation of IoT laboratories using artificial intelligence represents a novel approach aimed at streamlining the teaching process, providing immediate feedback, and minimizing the time spent on evaluation. This would allow instructors to focus their efforts on teaching itself. This chapter presents a comprehensive solution based on integrating sensor-based detection, advanced code analysis leveraging machine learning, and recommendations for improvements. The proposal comprises three main components: hardware analysis, code evaluation, and personalized feedback.

3.1 Hardware Analysis Through Image Processing

One of the central aspects of the proposed solution is detecting faulty connections in IoT device components. Analyzing physical configurations in real-time using cameras and image processing algorithms will be possible. The following steps are proposed:

1. **Data Collection:** The laboratory will have high-quality cameras to capture hardware connections. Each configuration will be recorded and stored in a database.
2. **Model Training:** Convolutional neural networks (CNNs) will be trained on a dataset containing images of both correct and faulty configurations. The dataset will be created by combining manually annotated images with synthetically generated data.
3. **Connection Analysis:** Once deployed in the laboratory, the AI system will compare real-time images with schematics of the intended configuration. It will identify issues such as incorrect polarity, improper GPIO pin connections, or missing components. The results of the analysis will be displayed to the student, along with recommended steps for corrections.

3.2 Code Evaluation Using Artificial Intelligence

A key element of the proposal is the automated analysis of IoT device code. This process will include syntactic, logical, and optimization assessments. The following approach is suggested:

1. **Syntactic Analysis:** Tools such as Pylint, PlatformIO, Cppcheck, SonarQube (with extensions for C/C++), Arduino Linter, and others will be integrated into the evaluation system to identify syntax and formatting errors. AI models based on the Transformer architecture, such as GPT (Generative Pre-trained Transformer), will be enhanced to detect advanced issues.
2. **Logical Analysis:** Machine learning models will be trained to identify common logical errors in IoT algorithms. For instance, the AI will highlight inefficient or incorrect implementations in energy management or sensor handling.
3. **Code Optimization:** The AI system will suggest improvements to existing implementations, such as optimizing loops, enhancing memory efficiency, or simplifying algorithms.

3.3 Personalized Feedback for Students

Personalized feedback will be essential for improving student learning. Our proposal includes the following components:

1. **Adaptive Recommendations:** Based on errors identified by the AI system, students will receive personalized recommendations considering their previous

mistakes and progress. For example, if a student frequently makes errors while working with GPIO pins, the AI will provide supplementary learning materials and examples.

2. **Tools for Program Analysis, Simulation, and Visualization:** Specialized tools like Python Tutor, Wokwi, and PlatformIO Debugger will help students better understand their code's functionality. Python Tutor enables step-by-step program flow visualization for languages such as Python, C++, Java, and JavaScript, displaying variable states and the call stack. Wokwi offers simulation of Arduino projects with virtual hardware, allowing students to observe program behavior when interacting with sensors, LEDs, and buttons. PlatformIO Debugger supports code step-through, breakpoint setting, and variable analysis, helping to identify syntax and logical errors in programs for Arduino and other microcontrollers. These tools combine visualization, simulation, and debugging to enhance understanding of programming concepts.
3. **Interactive Chatbot:** An AI chatbot integrated into the laboratory environment will answer students' specific questions in real time, encouraging them to learn problem-solving skills independently.

4 Integration into the Laboratory Environment

The proposed system will be integrated into the existing IoT laboratory infrastructure. Deployment will require:

1. **Hardware Support:** Cameras, sensors, and server infrastructure to process image and physical data and analyze code.
2. **Software Architecture:** A web interface for instructors and students where analyses, visualizations, and recommendations will be available.
3. **Security Mechanisms:** Protecting students' data and ensuring data integrity.

The proposed solution will be modular and scalable, allowing for gradual implementation and expansion of its functionalities.

5 Implementation and Experimental Testing

Implementing the proposed system will require a comprehensive approach, encompassing technical, organizational, and pedagogical aspects. This process will be essential for integrating the solution into the existing educational environment and ensuring its functionality under various conditions.

6 Infrastructure and Resource Preparation

The first step in implementation will involve a detailed preparation of the laboratory infrastructure. This phase will include securing hardware, installing software, and creating conditions for training AI models. Placement plans for hardware components such as cameras, microcontrollers, and sensors will be developed to ensure efficient data collection and generation. High-resolution cameras will be configured to capture detailed hardware connections, while servers will be optimized for processing images and analyzing large data volumes.

Software preparation will involve installing libraries for image processing (OpenCV) and implementing machine learning frameworks such as TensorFlow. This process will also include designing and initializing a database to store information on configurations and evaluations.

These steps will ensure the integrity and availability of data for further processing and visualization.

AI model training will focus on detecting hardware errors and analyzing code. Convolutional neural networks (CNNs) will be trained to recognize correct and incorrect configurations, while transformers will be used for code analysis and evaluation. An important part of this preparation will be creating datasets comprising manually annotated data and synthetically generated scenarios.

7 Deployment in Real Laboratory Environments

After the infrastructure has been thoroughly prepared, the system will be deployed in actual laboratory conditions. This step will include installing hardware devices in strategic locations and configuring the software environment to make it ready for use. Camera calibration will be critical to ensure the quality of image data, which AI models will subsequently process. Microcontrollers and sensors will be tested in simulations mimicking real IoT scenarios to ensure proper functionality and system integration.

The software layer's connection to the user interface will allow students and instructors to access analysis results, visualizations, and recommendations. The user interface will be designed to be intuitive, provide clear information, and minimize the time required for orientation within the system. A scalability module will also be implemented, enabling the system to expand to additional classrooms or educational institutions.

Users will be trained to ensure a smooth transition into practical use. The sessions will cover operating the system and interpreting the analysis results. This process will ensure a proper understanding of the system and lay the groundwork for its long-term and effective utilization.

8 Methodology of Experimental Testing

Testing will play a crucial role in evaluating the effectiveness of the implementation. Our experimental testing will focus on three main areas: verifying AI models' accuracy, assessing the feedback's efficiency, and analyzing the impact on educational outcomes.

The accuracy of the AI model will be validated using a test set of configurations and codes that includes representative samples of student work. The models will be evaluated based on accuracy, precision, and recall metrics, providing a comprehensive overview of their capabilities.

The evaluation of feedback efficiency will focus on the speed and quality of the information provided to students. Feedback will be reviewed by both students and instructors, ensuring a two-way validation of its effectiveness.

The impact on educational outcomes will be tracked by comparing groups of students who use the system with those working through traditional methods. Key metrics will include task success rates, improvements in practical skills, and student satisfaction.

9 Financial and Organizational Aspects

Implementing an AI-based assessment system in IoT laboratories requires significant financial and organizational investments. Hardware costs include high-quality cameras, computing servers, and IoT devices, amounting to approximately 15,000 euros per laboratory. Additionally, the development and training of AI models demand computational resources and qualified personnel.

From an organizational perspective, training educators to use the system effectively is essential. Integrating AI tools into the assessment process necessitates curriculum modifications and institutional support to ensure smooth adoption. Another challenge is continuously updating the dataset to enhance the accuracy of the AI system.

10 Expected Results and Benefits

We anticipate that implementing the system will yield the following results:

- **Error Detection Accuracy:** The system is expected to achieve significantly higher accuracy in identifying incorrect configurations and code. Unlike humans, who may overlook certain mistakes due to "author blindness," fatigue, or other factors, automated evaluation will consistently detect errors without these limitations.
- **Feedback Efficiency:** Students will receive immediate and personalized recommendations, accelerating their learning process.
- **Simplification of Instructor Work:** Automation will reduce the need for manual evaluation, allowing instructors to focus more on providing individualized support to students.

11 Ethical Considerations and Data Security

The application of AI in education raises important ethical and security concerns. Protecting student privacy is crucial, as AI-based systems collect and analyze personal performance data. Compliance with GDPR and similar regulations is necessary to ensure data security.

Another challenge is maintaining fairness in automated assessments. AI models must be carefully trained to avoid biases that could disadvantage certain student groups. Transparency in AI decisions and oversight by educators remain critical to ensuring equitable evaluations.

12 Future Perspectives

Based on the gathered results, the system's functionality could be expanded. Advanced artificial intelligence models could predict potential errors, helping students proactively prevent mistakes. Integrating augmented reality (AR) technologies could allow students to visualize configurations and their correct setups in real time. Such innovations could significantly enrich the educational process and further enhance its efficiency.

13 Conclusion

The automation of IoT laboratories introduces a new dimension to education, significantly enhancing the efficiency and quality of teaching. Integrating sensor-based detection, AI-driven code analysis, and other advanced technologies will enable students to understand the complex concepts of IoT better and develop interdisciplinary skills. Such a system will provide immediate feedback and tools for error identification and correction, accelerating their progress.

However, adopting this approach brings specific challenges. The reliability of automated systems will largely depend on the quality of training data, which must cover a broad spectrum of actual and potential errors. Additionally, investment in hardware infrastructure will be necessary to ensure accurate detection and smooth analysis. The effective deployment of such solutions will require experts in artificial intelligence, IoT, and education.

In the long term, the implementation of automated laboratories could not only improve student success rates and relieve instructors from routine assessment tasks. This would allow them to dedicate more time to individual student support and developing advanced projects. Systems designed to be intuitive and user-friendly will also lower barriers to adoption in educational settings.

Future research should focus on tailoring systems to the specific needs of individual educational institutions. Further development could include using augmented reality (AR) to visualize real-world configurations directly in a digital environment or implementing voice assistants to provide students with dynamic instructions and feedback during their work.

Lastly, ethical and security concerns arising from using artificial intelligence in education must be addressed. Ensuring personal data protection and guaranteeing that these systems complement rather than replace human teaching should remain a priority. If these challenges are managed successfully, automated IoT laboratories can become a cornerstone of modern educational practices, equipping students to meet future technological challenges.

14 Resources

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Baraniak, E. (2024). Prípadová štúdia - defektoskopia: Brain:it.sk. *Brain*. https://brainit.sk/pripadove-studie/defektoskopia

Benjamin. (2024). AI v laboratórnom hodnotení: Presnosť a rýchlosť na novej úrovni. *Slovensko.AI*. https://www.slovensko.ai/ai-v-laboratornom-hodnoteni-presnost-a-rychlost-na-novej-urovni/

Essa, M. E. S. M., El-shafeey, A. M., Omar, A. H., Fathi, A. E., Abo El Maref, A. S., & Lotfy, J. V. W. (2023). Reliable integration of neural network and Internet of Things for forecasting, controlling, and monitoring of experimental building management system. *Sustainability*. https://doi.org/10.3390/su15032168

Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning.* Center for Curriculum Redesign.

Nama, P. (2024). Integrating AI in testing automation: Enhancing test coverage and predictive analysis for improved software quality. *World Journal of Advanced Engineering Technology and Sciences*. https://doi.org/10.30574/wjaets.2024.13.1.0486

Nikolaidis, N., Flamos, K., Gulati, K., Feitosa, D., Ampatzoglou, A., & Chatzigeorgiou, A. (2024). A comparison of the effectiveness of ChatGPT and Co-Pilot for generating quality Python code solutions. *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering - Companion (SANER-C)*, 93–101. https://doi.org/10.1109/SANER-C62648.2024.00018

Owan, V. J., Abang, K. B., Idika, D., Etta, E. O., & Bassey, B. (2023). Exploring the potential of artificial intelligence tools in educational measurement and assessment. *Eurasia*

Journal of Mathematics, Science and Technology Education. https://doi.org/10.29333/ejmste/13428

Singh, A. P., & Tomar, P. (2021). AI and IoT capabilities: Standards, procedures, applications, and protocols. *IoT in Practice*. https://doi.org/10.1016/B978-0-12-818576-6.00004-6

Xu, L. D., He, W., & Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753