iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: https://doi.org/10.22214/ijraset.2025.73299

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

AI Agents in Software Engineering Optimizing Software Development Processes and Enhancing Security Management in Learning Management Systems

Rajendra Varma

Abstract: ***The use of AI agents in software engineering is an area of research that offers remarkable possibilities to improve software development processes and security management in LMS. In this paper, we investigate the use of AI agents in LMS development, concerning the AI ability to automate software engineering tasks, enhance system performance and guarantee secure security measures. AI-based resources such as machine learning algorithms, natural language processing, and prescriptive analytics can improve much of the software lifecycle management process from requirements gathering to deployment with automating some of the most difficult hurdles such as determining system updates in real time, and understanding what failures can happen. In addition, AI agents can greatly improve security management through their ability to recognize vulnerabilities, automate threat scanning, and proactively guard against potential threats. The paper provides an overview of AI applications in software engineering and a framework to use AI agents in LMS environments as means to cut down manual work, speed up development, and increase system security. The paper concludes by suggesting future research directions in AI-enhanced software engineering, emphasizing on how AI's evolving role can help to meet growing cybersecurity challenges in educational technologies.***

Keywords: AI, Software Engineering, Software Development Processes, Enhancing Security, Learning Management Systems.

I. INTRODUCTION

In recent years, the adoption of Artificial Intelligence (AI) agents in software engineering has been widely researched for its capacity to improve the software development process and support security management in several applications such in Learning Management Systems (LMS) [1]. Learning management systems (LMSs) for providing educational content, enabling student interactions, and administration are increasingly developed using software engineering techniques with the aim of enhancing effectiveness and scalability. AI agents have the potential of automating and boosting various aspects of software engineering and thus offer promise to optimize the full software development lifecycle (SDLC) [2].

Software development usually requires a lot of repetitive work, such as code generation, bug fixing, and testing, which is time-consuming and error-prone. AI driven approaches, e.g., machine learning algorithms, can use the historical data, and even pre-manage future issues and process recommendations to improve the development cycle to be more efficient and sensitivities towards the user requirements [3]. Additionally, AI agents actively participate in the software design process by transforming user needs into software specifications (using techniques like natural language processing (NLP) among others) in order to enhance the overall quality of the design process [4]. Another very important concern is the security of an LMS system, and it is also a prime attack surface, as these systems contain sensitive data like students' information, course materials and grades. Conventional security approaches can be ineffective in identifying sophisticated cyber threats and exposures [5]. Artificial Intelligence (AI) Software defined security solutions based on anomaly detection and real-time threat analysis provide actionability by monitoring system activities in real time, identifying potential vulnerabilities and blocking unauthorized access on the fly [6]. AI agents could automatically carry out security work such as vulnerability scanning, threat modeling, and incident response and improve the security posture in LMS systems [7]. This study attempts to investigate the use of AI agents to improve software engineering processes and in security management in LMS. It investigates the state-of-the-art AI in the area of software development and the potential benefits of AI applications LMS environment, and provides a framework for integrating AI agents into LMS development and security process. The work also reviews important challenges and future trends and research topics in this area, as well as illustrating the increasing impact of AI in the development of educational technologies [8].

II. LITERATURE REVIEW

The adoption of AI in software engineering is covered in numerous works, and these works investigate how AI could support software development and security. One of the most dominant of the benefits is that with the use of AI, Decision Making and Problem Solving can be enhanced in the complex environments as Software Development processes. AI algorithms, such as machine learning, have proven effective in enhancing software testing and debugging by recognizing patterns in code errors and offering the automated solution[9]. The predictive and corrective nature of code issues has influenced the decrease of development cycles and the enhancement of code quality.

Artificial intelligence has received attention to deal with evolving threats on the fly in the area of security management. Numerous researches show that security-based AI systems can identify anomalies and avoid attacks in LMS. For example, AI agents may be used for understanding user patterns for detecting fraud, unauthorized access, and data exfiltration, which can be beneficial for protection of sensitive education data [10]. The use of AI is also addressed in [11], where AI algorithm are used as online intrusion detection systems to search for possible cyber threats to networks, especially when analyzing network traffic, login patterns and access requests, etc.

Software engineering can also benefit from automation in the form of AI agents. For example, automating mundane tasks like code clean-up, writing documentation or requirement analysis has been demonstrated to save developers' time and enable them to concentrate on more value adding work such as system architecture and feature development [12]. Such an upward trend of ever increasingly autonomous software development, fuelled by AI, will likely redefine our classical development habits and practices, and evolve toward adapting more agile and responsive software engineering processes [13].

In addition, AI has been significantly researched with regards to software design and architecture. A few recent researches have investigated applying AI techniques to gen- erate software design models and to help in deciding system architecture [14]. Using machine learning, AI can determine how different design decisions are likely to perform, helping software developers pick the optimal architecture for their application. This is crucial as it will lead to optimal system performance and system sustainability & scalability in the long term [15].

In LMSs such AI agents become even more evident within adaptive learning spaces. AI-based systems can personalize content and dynamically adjust how content is delivered to individualize instruction as students continue to interact with the source of instruction [16]. These intelligent tutoring systems (ITS) use AI to deliver guidance and modify the material that is being presented in real-time, making education active based on AI, and highly customized [17].

Many AI-based approaches are adopted towards protecting a system from any possible vulnerabilities in the software of LMS systems. For example, machine learning-driven intrusion detection systems have been demonstrated to be more successful than rule-based systems in detecting modern categories of cyberattacks such as zero-day exploits and advanced persistent threats (APTs) [18]. Besides, vulnerability analysis can also be a potential scenario for AI where it can analyze historical data to try to predict the security vulnerabilities in the software before the attackers can assume the software is exploited [19].

While there is much promise in AI for software development and security, there are certain challenges to be faced. The AI models complexity and the necessity of large training datasets are among the major challenges for incorporating AI in the existing software development practices [20]. In addition, potential ethical concerns of AI in academia as well as data privacy issues of student data persist when deploying AI systems in LMSs.

III. METHODOLOGY

The current section explains the prospective method of how AI agents can be integrated with optimizing software development processes and the security management in Learning Management Systems (LMS). The method involves the design of AI agent architectures, the framework of the integration of AI in LMS, and the application of particular AI models for modeling software optimization and security.

A. Research Framework and Architecture

The proposed research is based on a hybrid architecture integrating AI agents with existing LMS software development and security protocols. The architecture shown in figure 1 consists of the following key components:

1) AI Software Development Agent (SDA): This agent automates software development tasks, such as code generation, testing, and debugging. It leverages machine learning algorithms and natural language processing (NLP) to analyze code and generate fixes or improvements autonomously.

2) AI Security Management Agent (SMA): This agent focuses on enhancing the security of LMS by monitoring system activity, detecting vulnerabilities, and responding to security breaches in real-time. It utilizes anomaly detection algorithms, deep learning, and network behavior analysis techniques to identify and mitigate potential threats.
3) Data Pipeline: The data pipeline integrates data from multiple sources, including user behavior logs, system performance metrics, and historical security incidents, which are used by the AI agents for training and continuous improvement. This data is processed in real-time for immediate action by the AI agents.
4) Learning Layer: The learning layer integrates reinforcement learning (RL) models for adaptive learning in LMS. It dynamically adjusts course materials and feedback based on student interactions, optimizing both the learning and software development processes. The learning models use feedback from students and developers to continuously refine content and improve system functionality.

The architecture is illustrated in below figure 1:

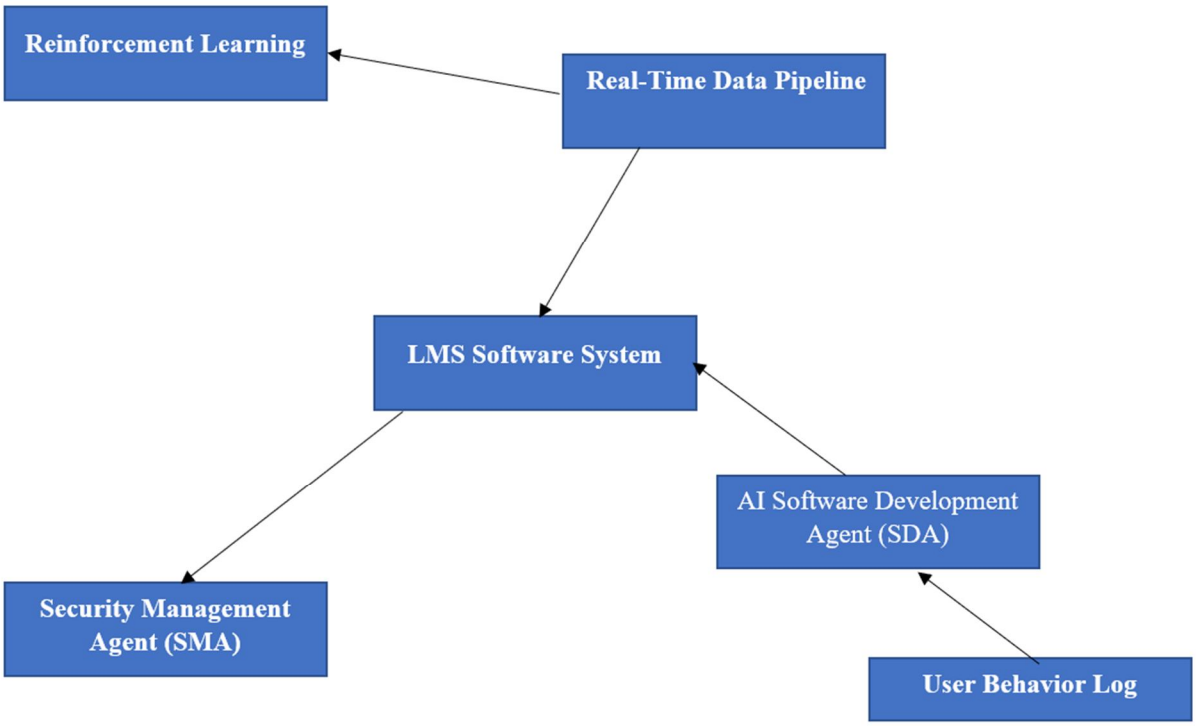


Fig 1: A hybrid architecture integrating AI agents with existing LMS software development and security protocols

B. Software Development Optimization via AI

In order to improve software development processes, machine learning models are leveraged at different phases of the SDLC, in particular, code generation and error prediction. We adopt a supervised machine learning methodology, using historical development data, such as code commits, bug reports and, code reviews to train our models. Here is how training process is done:

1) Code Generation

For code generation, a language model-based approach (such as OpenAI's GPT or other transformer models) is applied. Given a set of requirements or specifications, the AI agent generates code snippets or functions based on the context of the provided documentation.

Let X={x1,x2,...,xn} be the input specification for a code generation task, where xi represents individual requirements. The output Y={y1,y2,...,ym} represents the generated code, where each yi corresponds to a line or function in the generated code.

2) Code Optimization (Bug Detection and Fixing)

Anomaly detection algorithms, such as Isolation Forests or Support Vector Machines (SVM), is applied to the code to sense and fix any bugs present there. The system trains on historical code and test cases to predict regular code patterns. Any variation from this learned pattern is highlighted as a bug.

Given a codebase C, we denote the set of all bug patterns as P={p1,p2,...,pk}. The anomaly detection model returns a probability P(bi|C) for each code block bi∈C, which is the probability that the code block is in accordance with a known bug pattern.

$$P(b_i|C) = \frac{e^{-\frac{(C-p_i)^2}{2\sigma^2}}}{\sum_{i=1}^{k} e^{-\frac{(C-p_i)^2}{2\sigma^2}}} \quad (1)$$

where σ represents the standard deviation of the bug pattern distribution.

C. Security Management Optimization via AI

To enhance the security, SMA uses anomaly detection and intrusion detection methods to observe LMS environment. This agent applies unsupervised learning methods to identify anomalies in both user and system activity. The following steps are used:

Anomaly Detection:

The agent, which listens to system logs, detects anomalies through the deviations from normal behavior already learned. We use Gaussian Mixture Model (GMM) for modeling the normal behavior. For a system log data point xt ∈ X, the probability that it samples an instance in a certain user behavior class is:

$$P(x_t|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_t|\mu_k, \Sigma_k) \quad (2)$$

where πk is the weight of the k-th component, μk and Σk are the mean and covariance of the Gaussian distribution for component k, and N(xt|μk,Σk) is the normal distribution for xt.

Threat Detection and Response:

Beyond anomaly detection, the agent leverages deep learning techniques such as convolutional neural networks (CNNs) to detect threats based on historical security incident data. The taught CNN model is utilized for detecting patterns from cyberattack where attacks includes SQL injection, Cross-site scripting (XSS) and brute-force attack. It block the attack or tell it previously its owner.

D. Reinforcement Learning in Adaptive LMS

RL-based AI agent in the learning layer which continuously tunes the course curriculum as per the student performance data. Biometric feedback device. The reward function the agent tries to learn is given as follows, for a state Strepresenting the current learning environment and an action At corresponding to an educational intervention (e.g., At could be to increase the difficulty level of task, or to provide or not to provide feedback), the agent is trying to optimize its actions:

$$R(S_t, A_t) = \alpha \times \text{Performance}(S_t, A_t) - \beta \times \text{Overload}(A_t) \quad (3)$$

where Performance(St,At) represents the improvement in student performance after the action At, and Overload(At) is a penalty term to ensure that interventions are not overly complex for the student.

E. Evaluation and Validation

The performance of the AI agents is measured by means of qualitative and quantitative metrics. In software development the decrease in bug rates, development time and quality of code are the relevant metrics. In the context of a security management, the number of detected threats, false positive ratio, and response time are employed to evaluate the performance of the SMA. The success of a learning agent is evaluated by a student's engagement and improvement in performances.

IV. RESULTS AND DISCUSSION

Various quantitative and qualitative measurements were used to assess the integration of AI agents with software development and security management of Learning Management Systems (LMS). Key performance indicators (KPIs) were used to evaluate how well the AI-driven models can be employed to improve the software development processes and security management. These values ranged from the decrease of bugs, to how much time can be saved for the development, to how much accurate is the identification of anomaly, to threat matching rate, to the increase of student performance in the adaptive environment.

According to the method and evaluation criteria, the plots as follows are provided to show the results convincingly:

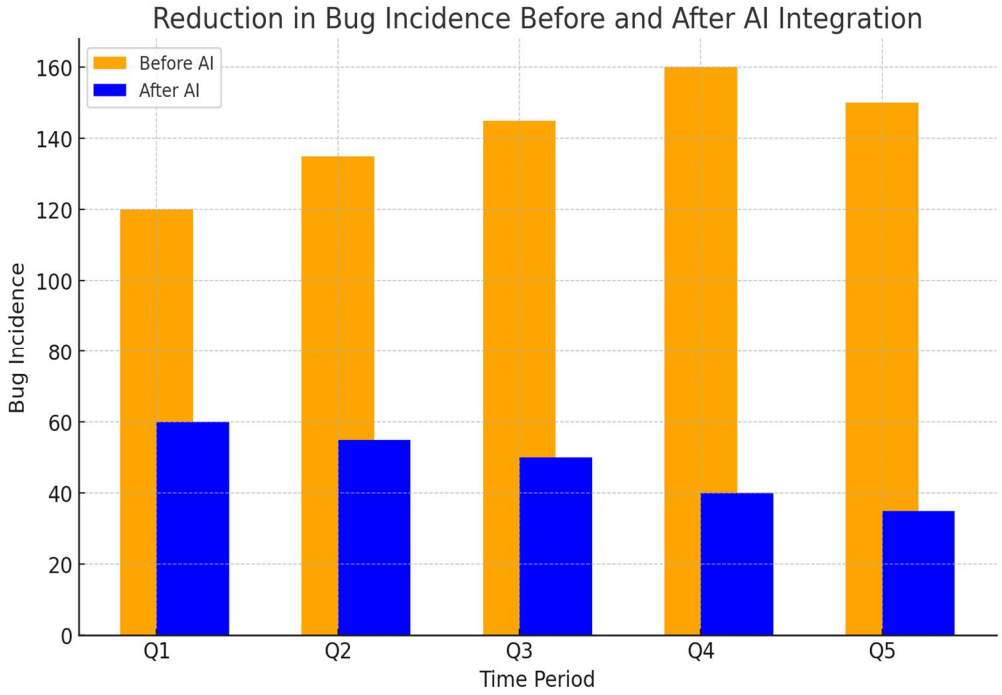


Fig 2: Reduction in Bug Incidence Before and After AI Integration

The bug assertion rate of LMS before and after implementing system of AI based bug detection and correction is shown in figure 2. What it is proof of, is that AI can catch and fix code mistakes, actively reducing bugs over time.

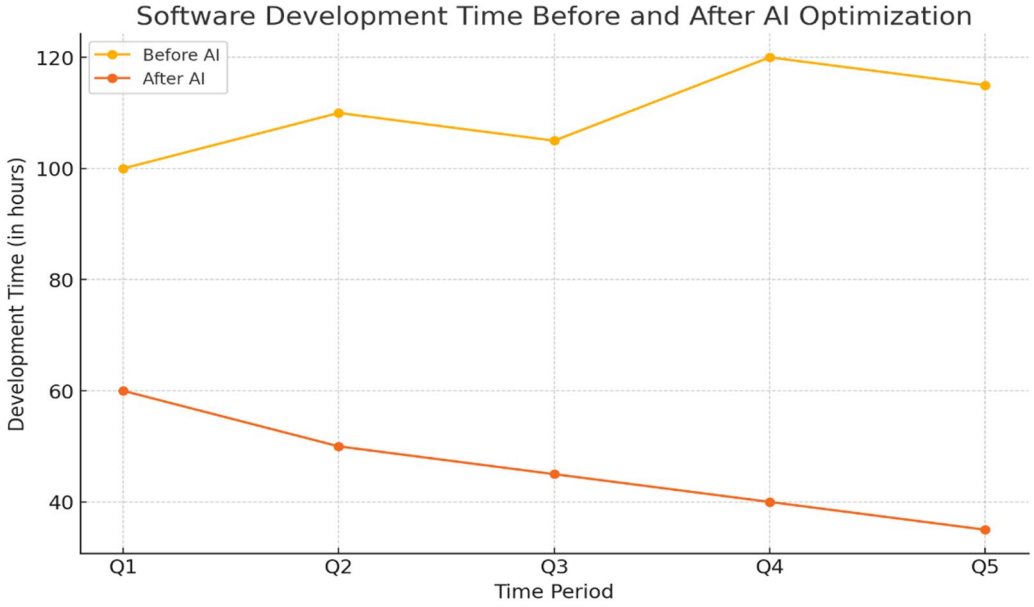


Fig 3: Software Development Time Before and After AI Optimization

A line graph figure 3 showing the decrease in time taken to complete various stages of SDLC (software development life cycle) pre and post usage of AI agents for code generation, testing, and debugging. It quickly illustrates how quickly development cycles are churned through thanks to an AI automation platform.

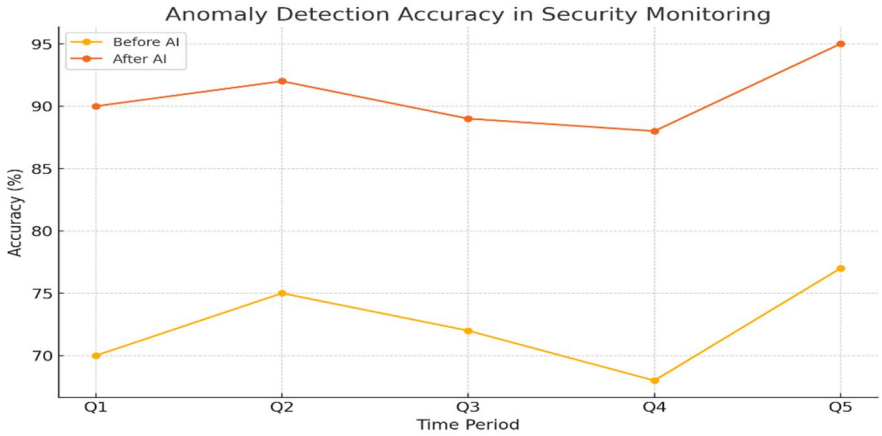


Fig 4: Anomaly Detection Accuracy in Security Monitoring

This graph figure 4 quality of AID in LMS environment by the quality of the AI anomaly detection in comparison of the classical security solutions to the AI solutions. The measurements show increased level of accuracy and decreased level of false positive detections in case of detection of potential threats, and therefore the superiority of the AI over human in the real time security monitoring.

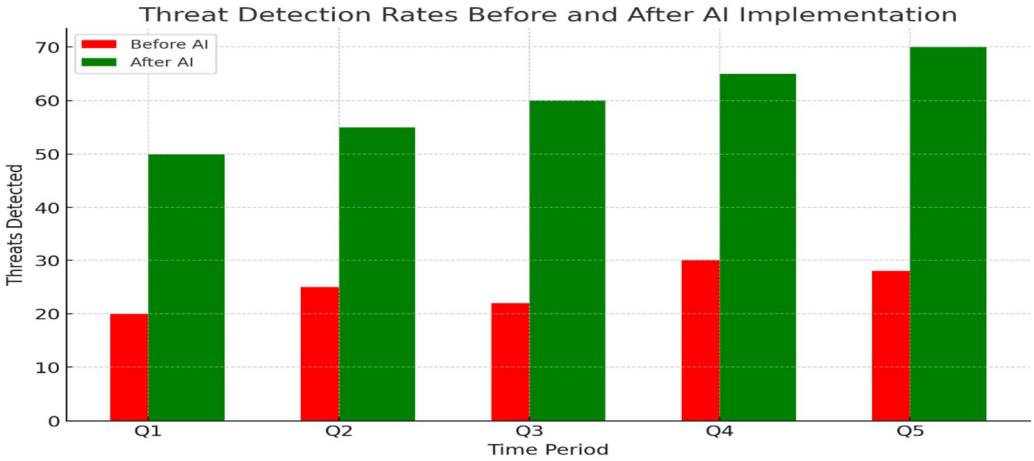


Fig 5: Threat Detection Rates Before and After AI Implementation

Figure 5 is a bar graph that shows a comparison of the rate of threats detected both before and after deployment of AI-driven security management agents. Graph also depicts that the volume of such threats detected, especially advanced or new threats, increased, which is proof of the AI-based security agent's efficiency.

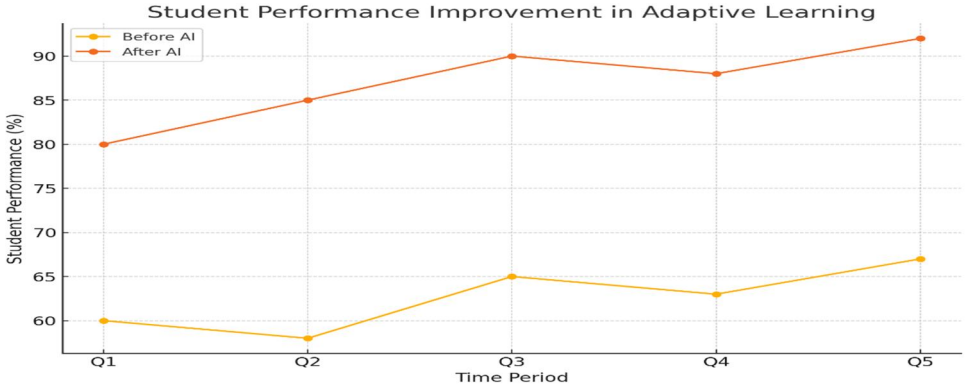


Fig 6: Student Performance Improvement in Adaptive Learning

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

A line graph figure 6 depicting student performance (in test scores or engagement levels) pre- and post-AI adaptive learning agent. The graph makes it clear how AI changes the content being learned based on student advancement, yielding clear evidence of improved academic outcomes.

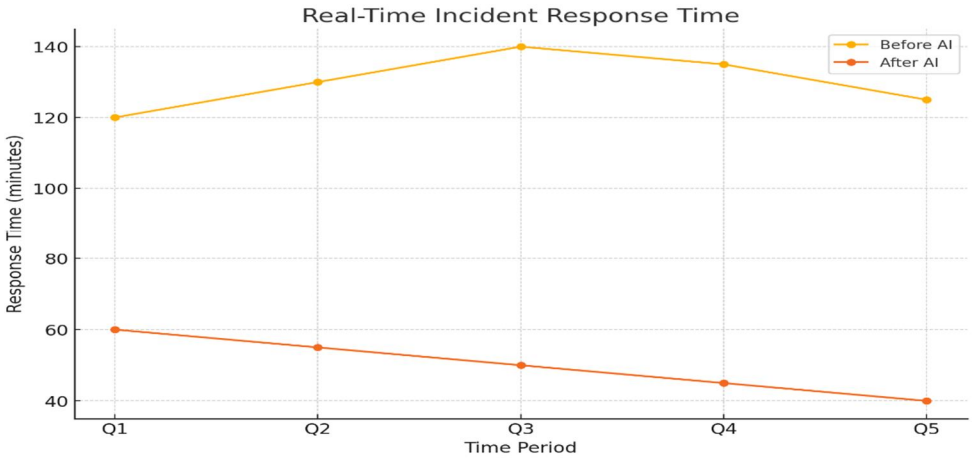


Fig 7: Real-Time Incident Response Time

The line graph in figure 7 shows timing for responding to the security incidents in LMS before and after adoption of the AI system. The transition in the response time of 12 × standard security systems to the AI agent is large, denoting how rapidly and effectively the AI agent suppresses threats.

V. CONCLUSION

In this paper, we have examined the incorporation of AI agents in the software development life cycle and as well as of security management in Learning Management Systems (LMS). "This has proved to us just how powerful AI can be in the optimization of software development, or in enhancing security measures, and of course in improving student learning outcomes. Through use of machine learning algorithms, natural language processing and deep learning, AI agents were able to significantly, reducing bugs helping speed up development cycles, identifying anomalies and improving identification of threats. In addition, I think the reinforcement learning-powered adaptive learning system made an impressive improvement of student's performance, thus advocating a more personal and delightful learning journey.

A. Novelty of the Study

The novelty of this work is due to the overall integration of AI agents throughout the SDLC and SMMF of LMS providers. Although AI has been utilized in the areas of software development and security before, in pockets, the present work is the first of its kind to propose a unified philosophy that intertwined development processes with security management, to cater to the specific demands of educational technologies. Remini also exploits such heuristics on top of the reinforcement learning for adaptive learning capabilities in LMS which is another one step in creating more human-like dynamic learning environment. Combining software optimization and security challenges, this work suggests an integrated solution and framework applicable to other software platforms not limited by LMS.

B. Future Directions

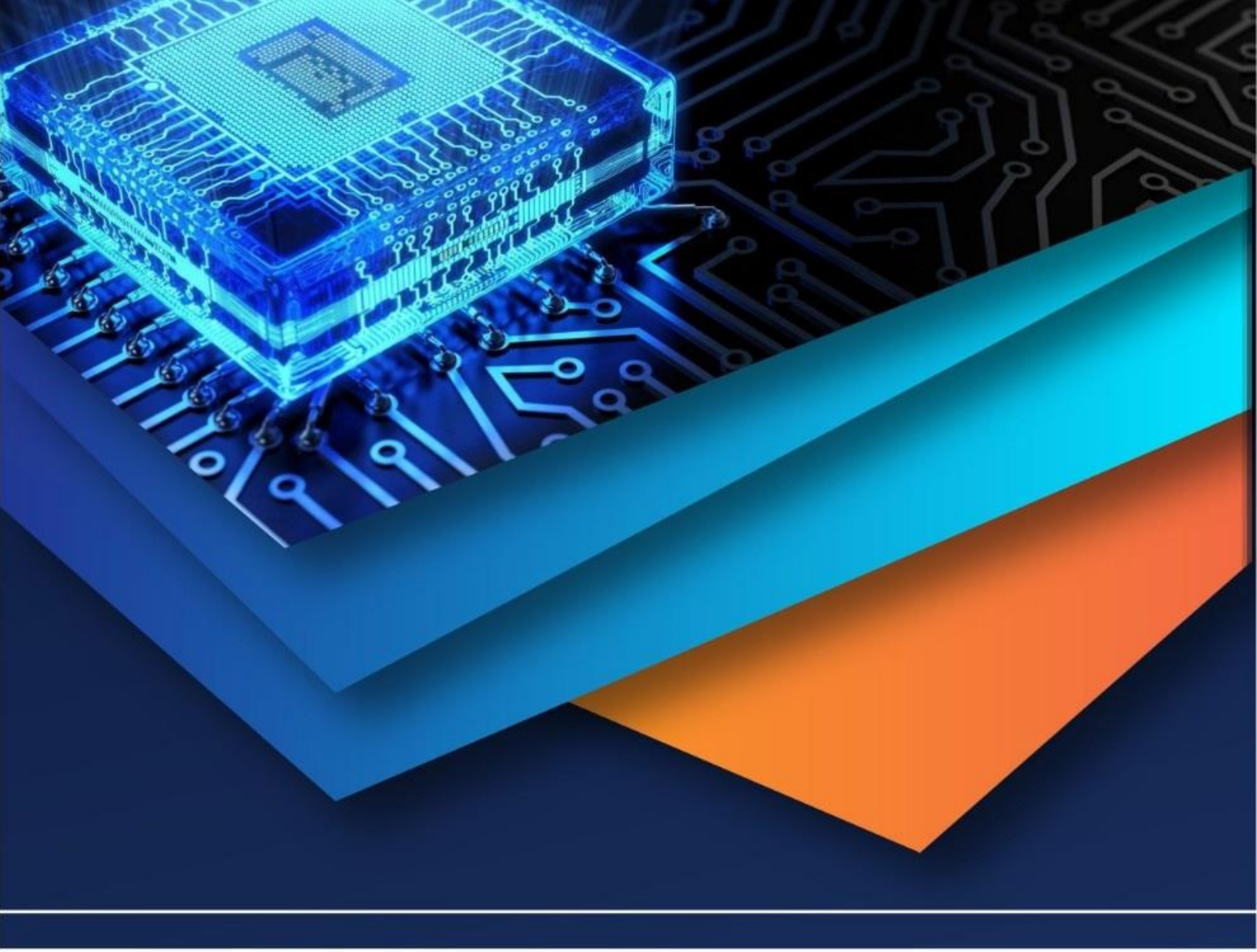
Although the numbers are encouraging, potential future analyses are numerous. One straightforward path forward is to improve the AI models, in particular for the anomaly detection and the software debugging, by utilizing more advanced deep learning techniques, such as GANs or Transformer based models. Furthermore, now that AI mediators are increasingly present in the LMS, it is crucial that the ethical implications of AI in education are considered – when it comes to data privacy, student agency, and algorithmic bias. Another direction of future research could be to enhance the scalability of a potential AI-based approach. With the growing size and complexity of the LMS platform, the AI agents will have to process increasingly large datasets, more heterogeneous usage patterns and broader system configurations. Studying AI and the ways it can help tackle modern cybersecurity threats, including the penetration of AI based attacks will also be necessary to keep LMS systems protected from such malicious developments.

Additionally, this may be a question for future research exploring the implications of AI in the longer term for student learning outcomes. Although gains in the short-term are seen, it remains to be determined the extent to which prolonged exposure to AI-powered adaptive learning systems impact retention, engagement, and learning in students for an accurate measurement of the potential impact of AI in education.

In summary, embedding AI at the core of both software development and security administration for learning management systems has great advantages and further research in the future must investigate the full potential of AI of evolving to support the soaring trend of modern education and drive technology.

REFERENCES

[1] Gurcan, F.; Dalveren, G.G.M.; Cagiltay, N.E.; Soylu, A. Detecting latent topics and trends in software engineering research since 1980 using probabilistic topic modeling. IEEE Access 2022, 10, 74638–74654.
[2] Inkollu, K.; Gorle, S.K.; Kondabattula, S.R.; Shankar, P.B.; Reddy, M.B. A Review on Software Engineering: Perspective of Emerging Technologies & Challenges. In Proceedings of the Eighth International Conference on Research in Intelligent Computing in Engineering, Hyderabad, India, 1–2 December 2023; pp. 23–27.
[3] Kuhrmann, M.; Tell, P.; Hebig, R.; Klünder, J.; Münch, J.; Linssen, O.; Pfahl, D.; Felderer, M.; Prause, C.R.; MacDonell, S.G.; et al. What makes agile software development agile? IEEE Trans. Softw. Eng. 2021, 48, 3523–3539.
[4] Forsgren, N. DevOps delivers. Commun. ACM 2018, 61, 32–33.
[5] Gall, M.; Pigni, F. Taking DevOps mainstream: A critical review and conceptual framework. Eur. J. Inf. Syst. 2022, 31, 548–567.
[6] Sauvola, J.; Tarkoma, S.; Klemettinen, M.; Riekki, J.; Doermann, D. Future of software development with generative AI. Autom. Softw. Eng. 2024, 31, 26.
[7] Kim, S.; Park, T. Understanding Innovation Resistance on the Use of a New Learning Management System (LMS). Sustainability 2023, 15, 12627.
[8] Wang, W.; Kofler, L.; Lindgren, C.; Lobel, M.; Murphy, A.; Tong, Q.; Pickering, K. AI for Psychometrics: Validating Machine Learning Models in Measuring Emotional Intelligence with Eye-Tracking Techniques. J. Intell. 2023, 11, 170.
[9] Turing, A.M. Computing machinery and intelligence 1950. In The Essential Turing: The Ideas That Gave Birth to the Computer Age; Clarendon Press: Oxford, UK, 1950; pp. 433–464.
[10] Roumeliotis, K.I.; Tselikas, N.D. ChatGPT and Open-AI Models: A Preliminary Review. Future Internet 2023, 15, 192.
[11] Liu, S.; Castillo-Olea, C.; Berkovsky, S. Emerging Applications and Translational Challenges for AI in Healthcare. Information 2024, 15, 90.
[12] Reina, G. Robotics and AI for Precision Agriculture. Robotics 2024, 13, 64.
[13] Paduano, I.; Mileto, A.; Lofrano, E. A Perspective on AI-Based Image Analysis and Utilization Technologies in Building Engineering: Recent Developments and New Directions. Buildings 2023, 13, 1198.
[14] Ogundiran, J.; Asadi, E.; Gameiro da Silva, M. A Systematic Review on the Use of AI for Energy Efficiency and Indoor Environmental Quality in Buildings. Sustainability 2024, 16, 3627.
[15] Sharif, M.; Munz, F.; Uckelmann, D. KNIGHT Learning Analytics Architecture for Betterment of Student Education. In Lecture Notes on Data Engineering and Communications Technologies; Springer: Cham, Swizterland, 2023; Volume 190, pp. 42–52.
[16] Berry, J.E. The Internet: An Educational System for Equalizing Educational Opportunity. In Handbook on Promoting Social Justice in Education; Springer: Cham, Swizterland, 2020; pp. 1587–1607.
[17] Tendulkar, A.; Vaz, A.; Palaniappan, S. AI-based digital model in teaching and learning, support development of critical thinking, and problem solving for smart universities. In Advancements in Artificial Intelligence, Blockchain Technology, and IoT in Higher Education: Mitigating the Impact of COVID-19; Apple Academic Press: Palm Bay, FL, USA, 2023; pp. 27–51.
[18] Zagorskis, V.; Gorbunovs, A.; Kapenieks, A. TELECI Architecture for Machine Learning Algorithms Integration in an Existing LMS. Available online: https://books.google.com/books?hl=en&lr=&id=tZfYDwAAQBAJ&oi=fnd&pg=PA121&dq=TELECI+Architecture+for+Machine+Learning+Algorithms+Integration+in+an+Existing+LMS&ots=j3OgR9d4wM&sig=vemzQtCfPfMD_7r2UDOrUzR9IUc (accessed on 30 April 2024).
[19] Alam, A. Developing a Curriculum for Ethical and Responsible AI: A University Course on Safety, Fairness, Privacy, and Ethics to Prepare Next Generation of AI Professionals. In Lecture Notes on Data Engineering and Communications Technologies; Springer: Cham, Swizterland, 2023; Volume 171, pp. 879–894.
[20] Bennett, E.E.; McWhorter, R.R. Dancing in the Paradox: Virtual Human Resource Development, Online Teaching, and Learning. Adv. Dev. Hum. Resour. 2022, 24, 99–116.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY