# The Role of AI in Automated Testing and Monitoring in SaaS Environments

**Ramesh Bishukarma**
**Independent Researcher**

**Abstract—** With the help of AI, automated testing and monitoring in the SaaS field is being improved significantly, becoming significantly smarter, faster, and more reliable. Incorporation of AI in SaaS platforms enables the platforms to diagnose problem areas, anticipate failures as well as enable efficient testing hence delivering optimal value and continuous service. This shift helps business by providing them with more intelligent and proactive ways to manage complex systems. The integration of Artificial Intelligence (AI) into Software as a Service (SaaS) environments has significantly transformed a landscape of software testing and monitoring. Automating testing procedures, improving efficiency, and minimising human work in Agile and DevOps approaches are all highlighted in this study as key roles of AI. Advanced AI leads to the testing capability of generating test cases automatically, executing these and analyzing performance, thus, identifying problems faster and more accurately. On the same note, AI improves the observation process by facilitating real-time observation of the performance of various systems and offering early management of disruptions. It is to understand that separate positive effects of AI implementation involve operation, expansion, and cost-effectiveness aspects that overall enhance the efficiency of SaaS applications. A review of the relevant literature underlines the progress in and further prospects for utilizing AI in enhancing testing and monitoring processes in SaaS contexts.

**Keywords**: Artificial Intelligence (AI), Automation, Test Engineering, automated testing, Test execution, SAAS, Cloud computing.

## I. INTRODUCTION

The rapid growth of SaaS environments has driven a need for more efficient, reliable, and scalable methods of software testing and monitoring. It has been realized that old-fashioned approaches to manual testing cannot meet the demands of today's complex Cloud environments. Bitter-sweet, artificial intelligence (AI) is an innovative game-changer that automates significant testing and monitoring functions and enhances both speed and precision significantly. AI in the form of ML, NLP, and predictive analysis is slowly being incorporated in to SaaS to detect errors, analyze consumption, and facilitate smooth operation of software [1].

Testing tools consisting of Artificial Intelligence facilitate creating test case, regression test, and performance monitor where the chance of human errors is minimized as well as the test time is minimal. Real-time analysis of enormous volumes of data, pattern recognition, and problem prediction are all made possible by these techniques. This strategic move does not only improve the credibility of SaaS applications but also reduces time loss and operation expenses. Therefore, AI automations have become crucial within a software development process and that is the proof that gives organizations an advantage in a world that is quickly turning into fully digital[2][3].

Apart from testing, AI has disrupted the monitoring in SaaS environments through performance measurement, and anomalous and anticipative monitoring. One way can use machine learning algorithms to always observe the overall health a system and automatically look at patterns that suggest that the system is degrading or is vulnerable. Due to the HIPs delivered by the implementations of the monitoring tools powered by AI, an organization is capable of delivering services to customers to the highest level while preventing any interruption that may lead to higher costs[4].

Through the incorporation of AI in SaaS environments the ways in which operations are conducted have been enhanced and the concept of business software testing and monitoring changed. Further developments in AI suggest that the future will bring full autonomy testing and visionary intelligent monitoring frameworks. These innovations will continue to enhance efficiency and capacity centrally relevant to SaaS application, paving the way towards next generation focus cloud solution [5].

### *A. Organization of the paper:*

The structure of the paper is as follows: Section II covers AI's role in automating testing in SaaS. Section III deals with the use of AI for supervision of SaaS environments with regard to performance measurement and fault identification. Section IV provides the benefits of AI integration and SaaS environment and Section V covers literature review on AI integration in SaaS for testing and monitoring, Section VI followed by the conclusion and future research directions.

## II. OVERVIEW OF ARTIFICIAL INTELLIGENCE IN AUTOMATION TESTING

Automated testing is currently a vital component of the development of software, especially in Agile and DevOps process models. It minimizes the amount of work done by hand, enhances test coverage and provides more accurate Testing process. New tests can be run frequently, thus increasing speed and efficiency in which, problems can be detected and bugs corrected to speed up the release of the software [6].

### *A. Key Features of Automated Testing:*

- Test Automation Tools: Software testing tools which can be used to automate testing are Selenium, JUnit, Test Complete, and Appium are used for web and mobile applications.
- Reusability and Speed: Automated test scripts are also reusable, and the ability to run these tests much quicker than performing manual tests will provide quicker feedback and the capability to integrate continuously.
- Improved Accuracy and Scalability: Full automation prefers less human interjection, producing standard outcomes in the long run. It also works well for testing whole systems for intricate compatibility with other systems across different frameworks [7].
- Benefits of Automated Testing: Automated testing expands on the test suite increasing the capability to

perform more tests, diverse and elaborate in nature. It also makes for its lower total costs and integrates both CI & CD process streamlining the whole software development process into a better quality.

### *B. The Role of AI in Automated Testing*

The following key points highlight the role of AI in enhancing automated testing processes.

- Enhanced Automation and Efficiency: A number of software testing processes are carried out using AI techniques thus reducing the amount of time and energy required for testing.
- AI-Driven Test Case Generation: Eggplant AI and Test.ai are some of the AI tools that perform the generation of tests using AI algorithm.
- Automated Test Execution: Some tests can be undertaken by an AI tool since computers do not require the assistance of people to do things over and over again.
- AI-Based Result Analysis: AI helps in making interpretations of the tests performed as it gives quicker and more accurate results.
- Reduction of Repetitive Tasks: AI enables testers to concentrate on more intricate and important areas by automating monotonous and repetitive testing chores.
- Predictive Defect Identification: AI, especially machine learning, can examine vast amounts of test data to find trends and anticipate any flaws.
- Improved Accuracy and Efficiency: AI improves testing speed and accuracy by finding even small flaws early in the development cycle.
- Intelligent Requirement Analysis: Improved test coverage is one result of AI's assistance in deciphering complicated software requirements and translating them into practical test cases [8][9].

### *C. Components of test automation framework*

There are a number of key parts to a test automation framework that work together to simplify testing [10]:

- Environment to be tested: It is essential to check and verify all test-related tools, equipment, scripts, and processes while building a test automation framework to guarantee the system can reliably carry out the intended task.
- Validate supporting tools: There are a number of useful tools that must be considered while arranging a test in order to write test scripts.
- Drivers & Library: The WebDriver can only be communicated and controlled by certain browser drivers.
- Organize Framework: Selenium is a platform for automating the testing of web applications.
- Testing Team: An effective testing methodology, competent testing personnel, and efficient testing tools are the pillars upon which a successful test automation project rests.


Figure 1: Components of test automation

The Figure 1. outlines the structure of an automation testing framework. It shows how the environment to be tested is central, with supporting components like the testing team, organization frameworks, drivers, libraries, and tools branching out, emphasizing the systematic approach to automation testing.

### *D. Types of Automation framework*

In the several types of automation framework, including Keyword-driven, data-driven, hybrid, linear, and modular. Each framework offers different benefits, depending on testing requirements:

- Linear Scripting Framework: This framework is referred to as the "Record and Playback" framework because it uses a simulated record and playback mechanism to build test scripts.
- Modular Testing Framework: This framework breaks down the whole application into smaller modules, and test scripts are written for each of those parts separately.
- Keyword Driven Testing Framework: The foundation of this system rests on the actions or keywords that are used to execute functions or procedures. Tabular format in an Excel sheet stores the test script that is entirely based on keywords and actions; this approach is also called Table-driven testing.
- Data Driven Framework: The foundation of the Data Driven Test Automation architecture is the separation of test data and test scripts. Test data is kept in distinct files or formats, and test scripts are connected to and run against distinct sets of test data.
- Hybrid Driven Testing Framework: In order to provide a more effective testing environment, this framework combines two or more frameworks, each of which contributes something unique [10].

### *E. Advantages of using artificial intelligence in Automation testing.*

The advantages of AI in automation testing are as follows in given below:

- Enhanced Precision: Even the most seasoned testers may make mistakes while manually testing software often.
- Beyond Manual Testing's Restrictions: Testing web applications in a controlled environment is challenging for even the biggest software development and quality assurance companies.
- Benefits for developers and testers: Before sending the code to the QA team, the developers may use shared automated tests to find problems more quickly.
- Increasing the total number of tests conducted: Automated software testing allows for more thorough and extensive testing.
- Time and money saved equates to a quicker time to market: It could be costly and time-consuming to manually execute these checks every time the source code is modified in software [11].

## III. AI IN MONITORING SAAS ENVIRONMENTS

Cloud computing allows users to access scalable computing resources and information technology services via the creation of virtual private clouds using resources from public clouds. Software as a Service (SaaS) is one of the service delivery models that will transform the way software is developed, sold, purchased, and used. Software is offered as a service under this paradigm, allowing cloud users to access it from their web browser without worrying about deployment, installation, or

upkeep. SaaS programs are often referred to as hosted, on-demand, or web-based software. The application's performance, availability, and security are maintained by the cloud provider. Using a multitenant architecture, SaaS cloud computing provides thousands of consumers with the necessary application over the internet[12][13].


Figure 2: SAAS architecture.

The Figure 2 represents a SaaS (Software as a Service) platform, illustrated by a laptop with 'SAAS' on its screen surrounded by cloud symbols linked to various elements such as code, databases, and global networking, indicating the diverse components and services offered through cloud computing.

### *A. Benefits of monitoring in SAAS*

In the following benefits of monitoring in SAAS include enhanced performance tracking, issue detection, and system health insights, which collectively improve service reliability and customer satisfaction are as below.

- It helps in enhancing the security of cloud applications and networks and ensures that there are no further security breaches.
- It also aids in easing the deployment of continuity plans. It strives to enable a proactive strategy rather than a reactive one for remediating risks and issues at the earliest.
- It leads to optimized service availability with instant issue reporting and fast solutions.
- As it is a subscription-based solution it keeps your cost relatively low and leads to minimize cost leaks.
- Additionally, you can see that scaling for increasing activity is smooth and effective for businesses of all sizes.
- It has an omnichannel network. Here, these tools can be accessed across several types of devices, which include computers, tablets, and phones. With this organization can track apps from any location too.
- It leads to simple installation because all of the requisites, such as infrastructure and configurations are already in place.

### *B. Security issues in SaaS*

The main security concern with SaaS is that the customer must rely on the supplier for all security precautions. A few security considerations that are crucial to the creation and implementation of the SAAS application service procedure [14][15]:

- Data access: Problems with accessing data stored in the cloud are associated with data access concerns. It has to do with different security guidelines that users may access while using cloud data. Security rules vary from cloud provider to cloud provider.
- Network security: Network security involves protecting any private information that travels across the network in order to stop unwanted access to sensitive data.
- Data integrity: The most crucial component of every system is data integrity. With a single database, data integrity may be readily attained in a standalone or single system.
- Data security: The sensitive data of any organisation is always located within the enterprise border in a typical application deployment architecture, and it is related to the people security, logical, physical, and access control processes.
- Data locality: Customers utilize the SaaS-provided apps to process their company data in a cloud environment using the SaaS paradigm.

### *C. Role of AI in SAAS*

The past five years have seen a meteoric rise in the impact of AI on SaaS product development. Advantages in cost, usability, and company scalability are offered by SaaS, a novel cloud-hosted software delivery paradigm. The fledgling Al SaaS is developing more quickly than previously thought. Its enormous potential is being recognised by organisations. Artificial intelligence is used by SaaS systems to provide individualised services, expand functionality, and improve performance, all while guaranteeing a rich user experience based on wise data-driven choices[16], [17]. The key points of AI in SaaS that are given below:

1. **Customer Support:** Customer service is essential for businesses of all sizes and in all sectors, and AI SaaS empowers it [18]. AI-powered automated chatbots simplify SaaS operations by quickly helping customers and pointing them in the direction of the best solutions without the need for human participation [19].
2. **Efficiency:** Business efficiency is enhanced by the optimisation of procedures made possible by artificial intelligence. Thus, benefiting productivity, or as in Al, automation drives away monotonous tasks.
3. **Enhanced Functionality:** Use AI in SaaS development to create user experiences that are flexible. Adapt app interactions dynamically by integrating powerful capabilities to analyse the behaviour of your target audience.

## IV. BENEFITS OF AI INTEGRATION AND SAAS ENVIRONMENT

An integration of AI into Software as a Service (SaaS) environments has revolutionized testing and monitoring processes. The application of AI improves efficiency, scalability, flexibility and performances within a business. This section aims to discuss with much specificity how best it may be used in a SaaS setting for increased testing and monitoring and in order to best utilize limited resources[20].

### *A. Improved Efficiency in Testing and Monitoring:*

AI replicates all the repetitive testing and monitoring procedures and makes it easier to detect problems. It is easier to diagnose bugs by using Algorithms and monitoring the execution of a program in real time thus enabling production of more reliable software with least time wastage [21].

### *B. Scalability and Flexibility in Cloud-Based Systems*

AI allows SaaS platforms to grow at a much more manageable rate to accommodate the increasing data usage and interconnectivity. It self-scales the cloud resources to utilize the available capacities proportional to the usage rates so that resources needed for high loads are instantly available [2][22].

### *C. Cost Reduction and Resource Optimization*

Lower operational costs are obtained by arising set of needs for labor involvement because AI is aimed at automating as

many methods as possible. The utilisation of cloud resources is also improved; when these resources are either overused or underutilised, it becomes quite expensive to get the most out of them [23][24][25].

## V. LITERATURE REVIEW

This section provides a literature review of AI's use in automation testing and monitoring in SaaS according to the following subtopics.

This study, Lokawati and Widyani, (2019) provides a solution to meet the needs of both service providers and renters by way of a monitoring system for SaaS that serves multiple tenants. SaaS is a popular model for distributing software that consumers may access from anywhere. For the most part, SaaS is developed using a multi-tenant approach. A single instance of a sharing application may support several tenants under this architecture. The service provider operating in a multi-tenant model must keep tabs on their tenants and the state of their services if they want to keep the service quality high. Assisting with the procedure would be a monitoring system. Finding a general monitoring solution that can be tailored to keep tabs on certain SaaS applications would be ideal[26], .

This paper, Li et al., (2019), integrated devices in a maize-growing environment, with the goal of collecting growth-related characteristics using an AI-powered embedded vision system and then monitoring the process using a user-defined algorithm. An integrated device-based system for tracking plant development incorporates a typical process flow with five steps: filtering, colour space translation, image segmentation, morphological operations, and feature quantisation. Numerous algorithms are included in each phase of the process, and users are free to mix algorithms to get plant analysis data in various environments[27], .

This paper, King et al., (2019), discusses the industry panel's views, ideas, thoughts, tactics, directions, and lessons learnt while creating software testing systems that leverage AI, testing AI systems using techniques, and creating self-testing systems. The expert panel and the testing community poll both produced enlightening viewpoints on AIST in action[28], .

The paper, Tao, Gao and Wang, (2019), gives us insight into AI software testing for demands and new features. Furthermore, several testing methodologies are examined and the categories for testing AI software are offered. Analysis of criteria and evaluation of test quality are also shown. Moreover, a metamorphic testing approach is used to conduct a practical investigation into quality validation for an image recognition system. Study findings demonstrate the approach's viability and efficacy[29], .

In this paper, Abderrazzak and Ahmed, (2015) remain committed to enhancing our MDA Framework, which aims to provide a SaaS inside a cloud environment using a multiview model with Coloured Petri Nets. We will be concentrating on the third of the four modules that make up this framework; it is responsible for designing an enhanced PNML that is supported by a multiview system[30], .

This paper, (Ziani and AlShehri, 2015) intends to provide a new architecture that would enable cloud ERP tenants to modify the ERP's source codes according to their needs. They were motivated by SAP systems to learn more about ERP system architecture and to suggest an ERP application that runs on the web. They next demonstrated how we enhanced the suggested web-based ERP application to provide customisation in a multi-tenant environment. A solid architecture underpins the suggested framework, which allows tenants to modify the cloud ERP system in accordance with their needs without interfering with one another[31],

Table 1: Summary of above given related work with key focused areas

| ef. | Domain | Approach | Focus of Study | Outcomes/Benefits | Challenges | Future Work |
|---|---|---|---|---|---|---|
| [26] | Multi-tenant SaaS Monitoring | Development of a generic, configurable monitoring system | Supporting service providers and tenants in SaaS environments | Generic system configurable for different SaaS applications | Difficulty in scaling to large, diverse SaaS environments | Improvement in scalability and real-time analytics |
| [27] | Agriculture (Maize Growth) | Embedded devices with user-defined algorithms for monitoring | Monitoring plant growth using embedded vision systems | Flexible algorithm combinations for tailored plant analysis | High computational demands and hardware constraints | Expanding to more crops and optimizing hardware performance |
| [28] | AI and Software Testing | Industry panel review with expert perspectives | Use of AI in testing software and designing self-testing systems | Insightful perspectives on AI systems testing and best practices | Complexity in integrating AI for automated testing | Develop more standardized AI testing tools and frameworks |
| [29] | AI Software Testing | Metamorphic testing method for image recognition systems | Assessing test quality and criteria for AI software testing | Demonstrates feasibility and effectiveness of metamorphic testing | Limited scope to image recognition systems | Expand testing to other AI applications and models |
| [30] | SaaS in Cloud Environments | MDA framework using Colored Petri Nets | Generating SaaS from a multiview model in cloud environments | Focus on the 3rd module of the framework using extended PNML | Complex integration of multiview models with cloud environments | Extend framework to support more complex cloud services |
| [31] | Cloud ERP Systems | Customization framework | Supporting customization in | Proposes strong architecture for | Managing tenant-specific | Explore further automation in |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | inspired by SAP systems | multi-tenant ERP without interference | tenant-specific customizations | requirements without conflicts | ERP code customization |

## VI. CONCLUSION AND FUTURE WORK

In conclusion, an integration of AI in Software as a Service (SaaS) environments has significantly transformed the landscape of software testing and monitoring. The integration of AI into the solutions is capable of presenting improved efficiency and scalability, in addition to greater accuracy of the testing processes as well as early identification of application performance bottlenecks. Applying the machine learning and predictive analyses, the organisation can prevent extra time for equipment failure and enhance the customer satisfaction. With SaaS platforms becoming even more intricate in the future, AI will become a critical factor for the stability and expansion of the offered cloud-based applications, which will give businesses a competitive advantage.

As future work in this area, efforts should be directed to enhancing AI for complete autonomy in testing and monitoring of SaaS ecosystems. Even better performance is predicted by seven-year AI advances in predictive maintenance, advanced anomaly detection, and self-diagnosing and self-healing technology and applications. Further, the studies should consider the idea of creating more effective frameworks of AI to correspond to the tendencies of change in the SaaS environments and implement constant learning.

## REFERENCES


[1] R. Pakdel and J. Herbert, "Adaptive Cost Efficient Framework for Cloud-Based Machine Learning," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, 2017, pp. 155–160. doi: 10.1109/COMPSAC.2017.42.

[2] A. Bahga and V. K. Madisetti, "Synthetic Workload Generation for Cloud Computing Applications," *J. Softw. Eng. Appl.*, vol. 2011, no. July, pp. 396–410, 2011, doi: 10.4236/jsea.2011.47046.

[3] V. V Kumar, "An interactive product development model in remanufacturing environment : a chaos-based artificial bee colony approach," *Eng. Int.*, vol. 6, no. 2, pp. 211–222, 2014.

[4] V. Narayan, "THE ROLE OF AI IN SOFTWARE ENGINEERING," *Int. J. Tech. Res. Appl.*, vol. 6, no. 4, pp. 34–36, 2018.

[5] C. Hung, Y. Hu, and K. Li, "Auto-Scaling Model for Cloud Computing System Auto-Scaling Model for Cloud Computing System," *Int. J. Hybrid Inf. Technol.*, no. April, 2012.

[6] A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," in *Future of Software Engineering (FOSE '07)*, 2007, pp. 85–103. doi: 10.1109/FOSE.2007.25.

[7] D. Benavides, S. Segura, and A. Ruiz-cort, "Automated Analysis of Feature Models 20 Years Later : A Literature Review 6," *Jornadas Ing. del Softw. y Bases Datos*, 2010.

[8] M. J. Goswami, "Utilizing AI for Automated Vulnerability Assessment and Patch Management," no. November 2019, 2019.

[9] R. P. Vamsi Krishna Yarlagadda, "Secure Programming with SAS: Mitigating Risks and Protecting Data Integrity," *Eng. Int.*, vol. 6, no. 2, pp. 211–222, 2018, doi: 10.18034/ei.v7i2.711.

[10] M. N. Islam and S. M. K. Quadri, "Framework for automation of cloud-application testing using selenium (FACTS)," *Adv. Sci. Technol. Eng. Syst.*, 2020, doi: 10.25046/aj050129.

[11] D. S. Battina, "Artificial intelligence in software test automation: A systematic literature review," *J. Emerg. Technol. Innov. Res. (JETIR*, vol. 6, no. 12, pp. 181–192, 2019, doi: 10.5220/0009417801810192.

[12] . K. V. K. M. K., "SOFTWARE AS A SERVICE FOR EFFICIENT CLOUD COMPUTING," *Int. J. Res. Eng. Technol.*, 2014, doi: 10.15623/ijret.2014.0301028.

[13] S. G. Priya Pathak, Akansha Shrivastava, "A survey on various security issues in delay tolerant networks," *J Adv Shell Program.*, vol. 2, no. 2, pp. 12–18, 2015.

[14] K. Yadav and N. Agarwal, "Cloud Computing: A review paper on security issues in SAAS," *Int. J. Eng. Res. Technol.*, vol. 3, no. 5, pp. 1164–1167, 2014.

[15] V. Kumar, V. V. Kumar, N. Mishra, F. T. S. Chan, and B. Gnanasekar, "Warranty failure analysis in service supply Chain a multi-agent framework," in *SCMIS 2010 - Proceedings of 2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering*, 2010.

[16] A. Achache, A. Baaziz, and T. Sari, "The Impact of Data Mining and SaaS-Cloud Computing : A Review," vol. 3, no. 1, 2020.

[17] V. V Kumar, A. Sahoo, and F. W. Liou, "ScienceDirect ScienceDirect Cyber-enabled Product Lifecycle Management : A Multi-agent Framework Cyber-enabled Product a Lifecycle Management : A Multi-agent," *Procedia Manuf.*, vol. 39, no. 2019, pp. 123–131, 2020, doi: 10.1016/j.promfg.2020.01.247.

[18] S. Pandey, "TRANSFORMING PERFORMANCE MANAGEMENT THROUGH AI: ADVANCED FEEDBACK MECHANISMS, PREDICTIVE ANALYTICS, AND BIAS MITIGATION IN THE AGE OF WORKFORCE OPTIMIZATION," *IJBQEAMR*, vol. 6, no. 4, pp. 13–19, 2016.

[19] S. Rathi, V. S. Rajput, and S. Gupta, "An improved evolution based optimization algorithm originated from the concept of SFLA and simulated annealing," in *11th International Conference on Industrial and Information Systems, ICIIS 2016 - Conference Proceedings*, 2016. doi: 10.1109/ICIINFS.2016.8262933.

[20] S. Gupta, N. Agrawal, and S. Gupta, "A Review on Search Engine Optimization: Basics," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 5, pp. 381–390, 2016, doi: 10.14257/ijhit.2016.9.5.32.

[21] M. V When, "When and what to automate in software testing ? A multi-vocal literature review When and what to automate in software testing ? A multi-vocal literature review," pp. 92–117, 2016.

[22] S. G. Ankur Kushwaha, Priya Pathak, "Review of

optimize load balancing algorithms in cloud," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, pp. 1–9, 2016.

[23] M. W. Shaukat, S. M. Raza, D. Y. Suh, and J. Piran, "A Review of Machine Learning Algorithms for Cloud Computing Security," *MDPI*, pp. 1–25, 2020, doi: 10.3390/electronics9091379.

[24] V. V Kumar, M. Tripathi, S. K. Tyagi, S. K. Shukla, and M. K. Tiwari, "An integrated real time optimization approach (IRTO) for physical programming based redundancy allocation problem," *Proc. 3rd Int. Conf. Reliab. Saf. ...*, no. August, 2007.

[25] C. Madhavaram, H. K. Gollangi, S. Bauskar, and E. P. Galla, "Unveiling the Hidden Patterns : AI-Driven Innovations in Image Processing and Acoustic Signal Detection," *J. Recent Trends Comput. Sci. Eng.*, vol. 8, pp. 25–45, 2020.

[26] H. Lokawati and Y. Widyani, "Monitoring System of Multi-Tenant Software as a Service (SaaS)," in *Proceedings of 2019 International Conference on Data and Software Engineering, ICoDSE 2019*, 2019. doi: 10.1109/ICoDSE48700.2019.9092741.

[27] Q. Li, S. Liu, Y. Mu, and M. Shang, "Maize Growth Monitoring Based on Embedded Vision System," in *Proceedings - 2019 2nd International Conference on Safety Produce Informatization, IICSPI 2019*, 2019. doi: 10.1109/IICSPI48186.2019.9096002.

[28] T. M. King, J. Arbon, D. Santiago, D. Adamo, W. Chin, and R. Shanmugam, "AI for Testing Today and Tomorrow: Industry Perspectives," in *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, 2019, pp. 81–88. doi: 10.1109/AITest.2019.000-3.

[29] C. Tao, J. Gao, and T. Wang, "Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices," *IEEE Access*, vol. 7, pp. 120164–120175, 2019, doi: 10.1109/ACCESS.2019.2937107.

[30] Z. Abderrazzak and E. Ahmed, "Cloud SaaS using MDA approach on a multiview models generate a SaaS from a colored Petri Net using view PNML," in *2015 International Conference on Cloud Technologies and Applications (CloudTech)*, 2015, pp. 1–5. doi: 10.1109/CloudTech.2015.7336977.

[31] D. Ziani and A. AlShehri, "A new framework for customizing ERP systems in a multi tenant SaaS environment," in *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, 2015, pp. 1–7. doi: 10.1109/WSWAN.2015.7209089.