International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V71I6P108

Volume 71 Issue 6, 47-54, June 2023

Original Article

Performance Test Engineering Practice for Scaled Agile Framework Leveraging Machine Learning and Artificial Intelligence Techniques

Suresh Kannan Duraisamy[1], Bharath Kumar Maganti[2], Ravi Pulle[3]

[1]*IT Senior Manager Automation and Performance Engineering, CSAA Insurance Group, Georgia, U.S.A.*
[2]*School of Information Technology, University of the Cumberlands, Texas, U.S.A.*
[3]*Principal Member of Technical Staff, Salesforce Inc, California, U.S.A.*

Received: 02 May 2023 Revised: 01 June 2023 Accepted: 14 June 2023 Published: 30 June 2023

Abstract *- The Information Technology (IT) industry constantly faces intense demand to meet evolving business & Customer needs, Leading to a proliferation of technology transformation initiatives such as the Scaled Agile Framework (SAFe), Cloud, and Microservices architecture. While these advancements promote the overall time to market, performance testing and other support services face a persistent challenge in keeping up with the speed of testing delivery. This research paper delves into the current state of Performance testing practices, explores the challenges of incorporating it into the SAFe software delivery model, and proposes solutions using DevOps automation, machine learning (ML), and artificial intelligence (AI). The research further highlights the importance of ML models in predicting system performance to accelerate test delivery; this has been demonstrated with a training dataset in Amazon's cloud Machine Learning solution-Sagemaker. This research also yielded a design for a next-generation AI-based expert system that can predict the performance of a software system using ML models and Rule-based engines.*

Keywords *- AI & ML in Software Testing, DevOps Automation, Performance Testing, Scaled Agile Testing.*

1. Introduction

In today's fast-paced digital world, the performance of applications plays a critical role in the success of businesses across various industries. Whether it is the frenzy of Black Friday online shopping or scheduling a hospital appointment, users often encounter frustration when faced with slow-loading applications. Studies have consistently shown that even a few seconds of delay can significantly drop user engagement, conversion rates, sales, and brand reputation. Performance test engineering has evolved as a crucial segment in the software development process. It has matured in validating application performance through load simulation tests to determine scalability, breaking points, endurance, resiliency, and overall reliability. Performance Engineering is particularly critical in industries with high transactional volume or/. It has vital services such as e-commerce, banking, finance, ticketing, healthcare, and hospitality, where a seamless user experience is paramount.

Artificial Intelligence (AI) and Machine Learning (ML) are rapidly emerging in various fields, including Performance Engineering. The ability of AI-based systems to incorporate human intelligence into machines has appeared as a revolutionary solution for advancements in many industries. Machine learning, a subset of AI, uses past data trends to build patterns and predict outcomes. The ML model aims to improve the accuracy of outcomes by learning various datasets. AI/ML has accelerated growth by assisting businesses in making predictions, decision-making, and operations. These capabilities of AI/ML have vast potential in the Performance engineering space that will be discussed in this paper.

2. Current State Performance Engineering in Scaled Agile Software Delivery

The SAFe delivery model was designed to address the limitations of the Agile model to scale for large enterprise-level initiatives comprising larger teams in a distributed development environment with multiple complex application integrations. The Scaled Agile Framework (SAFe) is a system for scaling Agile across teams of teams, business units, and even entire organizations [1]. Performance test engineering is an integral part of the SAFe delivery and is often a shared support service not directly embedded into Scrum teams. An enterprise performance engineering team enables the performance test engineering needs of every SAFe Agile development team. In the SAFe ecosystem, due to the size of the programs and multiple application dependencies, performance testing necessitates greater

collaboration with the Architecture, Engineering, and Product teams from the inception of the software development life cycle to ensure Performance issues are addressed early in the life cycle to keep up with the speed of testing delivery expectations.

In partnership with the DevOps, SRE, and Engineering teams, the performance team enables a continuous testing pipeline in production-sized performance environments for the main application branches on a regular cadence to capture the system's overall performance issues [2]. Furthermore, the performance engineering team empowers the scrum teams with a shift-left performance testing capability to run unit testing at the feature branch level before merging the new development to the application main branch. This approach helps detect poor-performing code methods and SQL degradations much earlier in the development process, eliminating the need for late-stage identification and saving valuable time for the engineering teams to focus on system-level performance issues. Moreover, this approach also improves the overall cost of Quality (CoQ), as the early resolution of performance issues is substantially less expensive than addressing them later in the software development lifecycle.

3. Quantitative Analysis of Performance Engineering Phases in SAFe Delivery

A thorough quantitative analysis was conducted, surveying numerous Performance engineers and Architects who have successfully adapted the SAFe model within their organizations. The objective of the survey was to comprehend the time and effort invested in each of the crucial phases of performance testing, including NFR (Non-Functional Requirements) gathering, workload analysis, script development, environments, Test Data Management (TDM) load test execution, analysis & reporting. The findings, along with the identification of the most time-consuming phase, are summarized in Table 1.

Table 1. Performance Testing Time Estimate in Agile and SAFe

Performance Test activities	Agile (1 Sprint)	SAFe (1 Sprint)
NFR Gathering	4Hours	4Hours
Workload modeling	4Hours	4Hours
Script Development	8-16 Hours	8-16Hours
TDM	4-16Hours	4-16Hours
Environment Management	4-16Hours	8-16Hours
Test Execution	8-12Hours	8-12Hours
Analysis & Report	8Hours	8Hours
Total Time	40-60Hours	40-60 Hours

Most engineers in this survey rated script development as the top time-intensive phase. Environment stability, a pivotal factor, was regarded as an unpredictable, challenging effort given the extensive time required for troubleshooting and fixing issues to stabilize the environment before heading toward performance test executions. The survey report revealed time-consuming critical hotspots within the performance test lifecycle, as depicted in the pie chart below. This visibility presents valuable insights into potential areas for optimization to enhance the overall speed of testing delivery in the SAFe Development model.

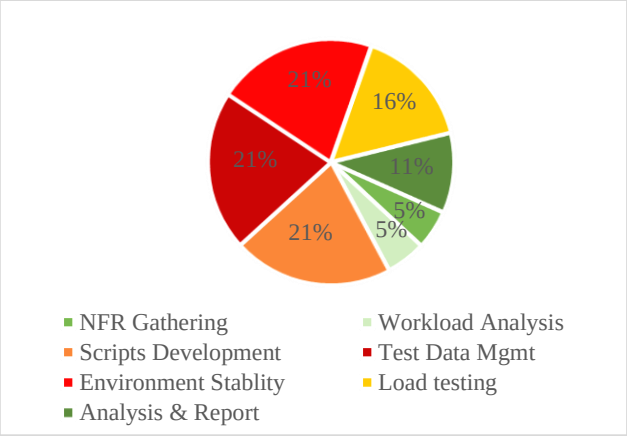

Fig. 1 Time consumed for each phase in the performance testing

4. Performance Testing Opportunities in SAFe Delivery

The pressing need for accelerated software product delivery poses a significant challenge for performance engineering practices in meeting the timely delivery of test findings within the fast-paced SAFe delivery model. The following sections discuss the substantial opportunities and potential solutions for the top time-consuming areas uncovered through the quantitative analysis.

4.1. Fulfilling the Demand for Faster Test Delivery

With the emergence of SAFe and Agile frameworks, the software development lifecycle has transformed into a nimble, incremental, and iterative model that has impacted every development support service, including performance testing. Performance engineering has evolved with DevOps-centric automation practices to meet this demand. DevOps's most significant aspects include a culture of collaboration, a focus on automation, clear metrics, and measurements[3]. This evolution in DevOps automation practice has enabled rapid software delivery with event-driven automated testing in Continuous Integration and Continuous Delivery (CICD) pipelines, as depicted in Figure 2.

The integration of Performance testing tools and test suites into CICD pipelines leads to a reduction in significant manual test execution efforts and, to a greater extent, can accelerate the overall test execution process. This can be implemented by configuring the check profiles into the Performance Test tool, which can signal the status of key

performance indicators as pass or fail in real-time into the CI-CD Pipeline[4]. The adoption of test integration in the CI-CD pipeline can undoubtedly increase the frequency of load tests and earliest reporting of degradations and enable resilient & reliable automated processes [5]

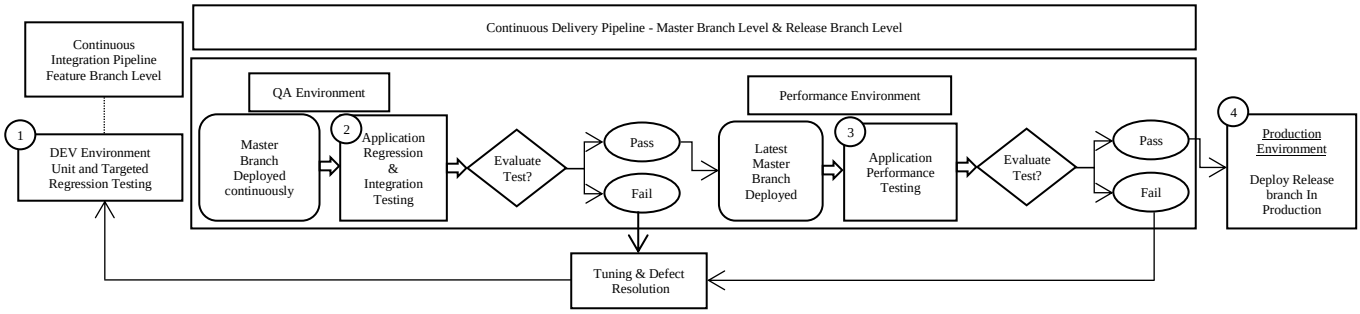

Fig. 2 Testing cycles automated in CI-CD pipeline

4.2. Operational Intelligence and OnDemand Provisioning of Performance Environments

Performance Environments are complex environments that are scaled like production capacity with end-to-end integration as much as possible; they demand regular updates and deployments to keep up with the latest application main and release branches, often leading to instability issues. These issues can manifest in various ways, including outages, deployment failures, functional errors, and integration issues, which cause significant impediments to the performance testing process and slow down the overall Product delivery.

The overall health status of the Performance Test Environment must be continuously monitored to ensure steady uptime. This is best achieved using cutting-edge telemetry tools, which can be deployed in application servers to monitor critical resource metrics. The telemetry tools continuously ping the servers to capture the systems availability and performance metrics instantaneously transferred to the Alert management system and CI-CD pipeline. This allows the performance testing automated jobs in the CICD pipeline to initiate or pause based on the current environment stability.

Another aspect of Performance Environment Management is the cost of operation to maintain such infrastructure, as it entails significant expenses due to the extensive production line capacity and the Licenses for the monitoring & Logging Tools. These expenses can be significantly curtailed by accurately ascertaining the required operational period that can activate on-demand environments. The utilization of Infrastructure-as-a-code tools, such as Terraform, enables environment management to be fully automated. A multi-cloud IaC (infrastructure as code) tool allows users to define a desirable infrastructure in a declarative language[6]. This cutting-edge feature can be seamlessly introduced as a self-service option, substantially reducing operational expenditure and eliminating the effort required to administer such environments. Furthermore, adopting custom instrumentation developed through open-source tools can replace costly monitoring tools and significantly save the overall cost of operations (TCO) compared to licensed Application Performance Management (APM) tools.

4.3. Automation of Test Data Management Process

One of the indispensable prerequisites for any performance test exercise is the availability of test data to support load testing and simulate a production-like workload. A large Volume of Data is needed quickly, and the required tools to extract the test data are often unavailable to performance test engineers [7]. Mining quality and manual test data practices further create bottlenecks in DevOps and CI/CD Pipelines, as testers save time creating or waiting for test data[8]. The creation of test data for complex integrated systems is often a time-consuming process, mainly when accomplished through front-end application simulation. Although an alternative mechanism to generate the required synthetic data using SQL scripts directly in the database is much faster, it fails to meet the speed requirements of testing in the SAFe model. An efficient approach to managing test data is to periodically monitor the volume of available test data and initiate the data creation process when the volume reaches a predefined threshold. This method fully automates test data management and enables the integration of performance testing into the continuous delivery pipeline.

Database volume is another critical aspect of the performance testing process, as maintaining an equivalent volume of data closer to production is essential to replicate production behavior and obtain reliable results[4]. Maintaining a copy of production data in a performance environment poses challenges, considering the time it takes to import a production-sized snapshot into a performance environment and data security compliance concerns. There are potential data migration tools with masking and encrypting capabilities, such as Actifio, that can securely

expedite the database loading process into the performance environment.

5. Application of AI & ML in Performance Testing

Artificial Intelligence is proven to have greater efficiency, improved productivity, and reliability. It has become an integral part of almost every Industry [9]. It is estimated that employing Artificial Intelligence in several functions could result in significant savings in operational expenditure across the globe [10]. AI Operations(AIOps) is about leveraging Artificial Intelligence in Information Technology operations which assists engineering teams in making better decisions and reducing manual effort. AIOps in IT have a wide range of use cases, from automating repetitive tasks to complex use cases such as managing failures in production systems. AIOps is intended to empower engineers to build software products efficiently and effectively with Artificial Intelligence and Machine Learning Techniques[11]. AIOps can boost business operations efficiency, reduce expenditure, and improve productivity when effectively developed and employed.

AIOps in performance engineering represent one of the largely untapped areas with huge potential. AI-enabled performance engineering can help organizations improve the performance and reliability of their software systems and reduce the costs associated with performance issues and downtime [12]. By integrating effective machine learning models, AIOps can be strategically leveraged to streamline performance test engineering processes that can empower early detection of Performance issues and increase productivity in record time. Rule-based models are the most effective for performance engineering applications among machine learning models. These models must be carefully selected and trained using vast data sets to maximize their efficacy. Doing so enables a wide range of capabilities, including Test development, Performance Analysis, Performance Prediction, and Capacity planning.

5.1. Test Script Development and Maintenance

Within the Performance Test Life cycle, the test development phase consumes about 27% of the time. Test scripts are developed by capturing the network protocol level traffic and enhancing it further to simulate the end user traffic and are one of the top time-consuming efforts in Performance testing [13]; the toilsome part of scripting is handling dynamic server responses in real-time with parameters so that the test scripts can run successfully for different data variables. AIOps can help significantly reduce this time and effort by employing a rule-based AI System. This model could build rules by traversing the application and capturing an index of dynamic values ingested into an AI system. The raw script automatically correlates all the dynamic values with necessary, updated object ids and uploads the scripts to the Load testing tool for the test to kick off without any overhead, as defined in the following process flow Figure 3.

This rule's engine could also restore test scripts that fail on a subsequent version of an application by comparing their status with what was previously documented [14]. This is one of the critical aspects of AI applications in performance test suite maintenance efforts. Rapid application development in the SAFe model causes frequent changes in application code, properties, and functional flows. It is challenging for the performance test team to keep the scripts updated with these changes, breaking the test scripts in the automation pipeline during runtime. AI help address this issue by providing a self-healing feature that scans the application to capture the latest parameters and object ids and updates the script with those changes in real time. This capability saves ample time for the testing teams and empowers the CICD pipeline to be more stable and robust.

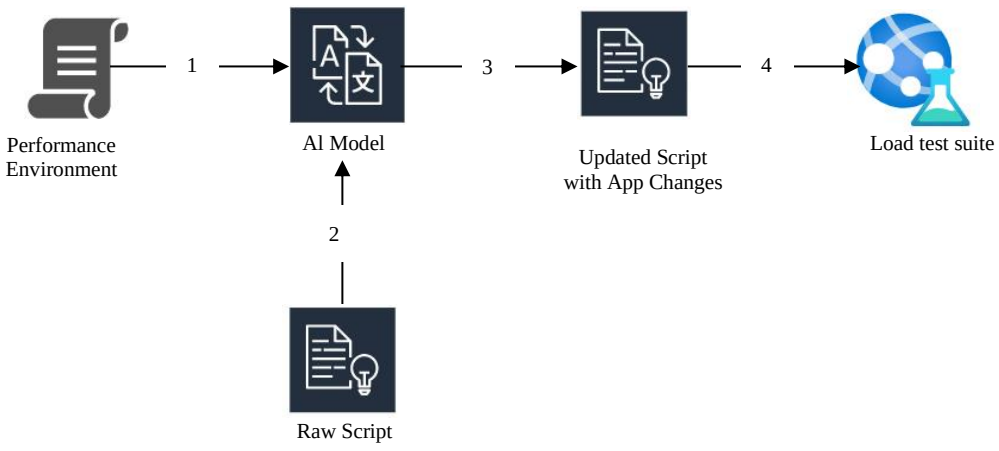

Fig. 3 Test script process flow through AI model

6. Importance of ML & AI to Accelerate the Speed of Testing Delivery

Performance Engineering is an important segment of the software development process that profoundly impacts business and customer experience. The rapid software development model, such as SAFe, presents a significant opportunity to expedite the speed of performance testing delivery. While DevOps automation techniques, CICD and Shift Left have significantly improved the efficiency and speed of software development and testing processes, the trajectory of future demands for accelerated performance testing seems unstoppable. Leveraging the capabilities of Artificial Intelligence (AI) and Machine Learning (ML) in performance testing can take it to a whole new level. AI and ML algorithms can analyze vast amounts of data generated from performance tests and identify patterns, anomalies, and correlations that may not be apparent manually. DevOps automation techniques provide continuous integration and delivery capabilities but may not provide real-time insights into application performance under various load conditions. AI and ML can monitor performance metrics in real-time, detect performance bottlenecks, and trigger automatic actions or alerts when certain thresholds are exceeded. This proactive approach enables faster identification and resolution of performance issues, minimizing their impact on the overall system.

6.1. Leveraging Machine Learning to Predict Application Performance

Machine learning solutions can analyze and interpret thousands of statistics per second, providing real-time (or near real-time) insight into a system's performance. They can be used to recognize data patterns, build statistical models, and make predictions invaluable to performance monitoring and testing [15]. The continuous load testing in the perf environments generates an astronomical amount of test result data, Server Logs, and server monitoring data. This dataset contains many features (inputs) with Test results and Application Performance outcomes; having a deeper insight into these datasets can lead us to understand the various input features and their relation. This can lead to Developing an efficient machine Learning model to predict the application performance over a period and any anomaly detections while taking prior testing into account [16].

The critical performance indicators, including user transactions, throughput (measured in transactions per second), and failure rates, can be ingested from the load test result file, as illustrated in the Table for a sample test execution. Crucial server performance indicators such as CPU usage, memory usage, and Disc I/O can also be leveraged for building a model. Understanding the input features' relationship is critical to developing the Machine Learning algorithm and model. The ML algorithms use the data sets' patterns to carry out future predictions and classification [17]. It can be a linear regression relationship between input parameters like response time deteriorates with an increase in Load. Defining a clear objective of the ML model is critical to accomplish the Outcome prediction based on the input feature parameters.

Table 2. Sample performance test result data table

Transaction Name	90% Latency SLA & Current		Count SLA & Current		Failures (%) SLA & Current	
Transaction01	5	0.24	100	115	1	0
Transaction02	5	0.17	100	114	1	0
Transaction03	5	0.23	100	114	1	0
Transaction04	5	2.17	100	114	1	0
Transaction05	5	0.06	100	116	1	0
Transaction06	5	1.56	100	116	1	1.7
Transaction07	5	0.6	100	116	1	0
Transaction08	5	0.11	100	115	1	0
Transaction09	5	0.14	100	115	1	0
Transaction10	5	1.06	100	116	1	0

The test result data sets must be continuously ingested into the machine learning algorithm to train and develop an efficient model with minimum loss and higher accuracy. The workflow in Figure 4 shows the approach for developing an efficient machine-learning model for Performance Prediction.

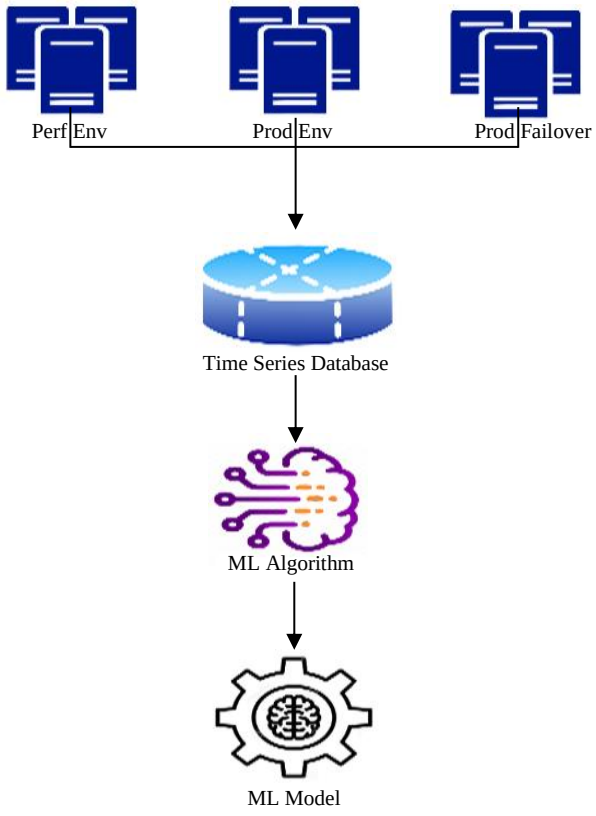

Fig. 4 Developing ML model for performance prediction

6.2. Demonstration of Performance Test ML model with AWS Sagemaker

The above concept has experimented with a cloud Machine learning platform called Amazon Sagemaker to build and test a sample ML Model. Amazon Sagemaker is a managed service in the Amazon Web Services (AWS) public cloud. It provides the tools to build, train and deploy machine learning (ML) models for predictive analytics applications [18]. Many such solutions with Prebuild Machine learning algorithms are available to build, train and test ML models quickly and efficiently. Performance test result data has been cleansed and uploaded to AWS -S3 bucket.AutoML mode was selected to build, train, and tune the best model based on uploaded Performance test data. There was numerous model that Sagemaker recommended after training the data, and the best one was selected to evaluate the model. The chosen ML model to predict the 90th percentile response time is as below

{
"label": "90th",
"headers": [
"90th",
"Transaction Name,"
"Min",
"Avg",
"Max",
"Completed",
"Success",
"Failure(%)",
"TPS",
"TPS",
"TPS(%)"
],
"dataset_type": "text/csv",
"methods": {
"shap": {
"baseline":
"",
"",
],
"num_samples": 2048,
"agg_method": "mean_abs",
"save_local_shap_values": true,
"use_logit": false
},
"report": {
"name": "explainability-report",
"title": "Explainability Report"
}
}
}

After the model was finalized, it was deployed to evaluate with new datasets excluding 90th percentile response time. For this experiment, the Batch prediction was exercised; however, there are options to make real-time predictions with Sagemaker endpoints also. The Batch job was executed, and the output file was generated with the 90th percentile response time prediction in the AWS S3 Bucket after about 8 mins. The 90th percentile predictions in the output file were very close to the actual result, thus ensuring the reliability of the Prediction. Sample transactions from the top of the list with actual and Predicted results are displayed in the following Table.

Table 3. Predicted outcome of performance test result data set

Transaction Names	90 % Response time	
	Actual	Predicted
Transaction1	0.24	0.316215
Transaction2	0.17	0.140684
Transaction3	0.23	0.253965
Transaction4	2.17	2.245457
Transaction5	0.06	0.126709
Transaction6	1.56	1.613023
Transaction7	0.6	0.603624
Transaction8	0.11	0.128626
Transaction9	0.14	0.16884
Transaction10	1.06	1.121563

The ML model developed in this experiment aims to estimate and predict the outcome for any missing Labels in the newly ingested data sets continuously flowing into the ML model. Expanding on such an ML platform and defining more objectives would uncover many hotspots, Anomalies and enable teams to proactively identify and resolve critical performance issues in a complex distributed system. Once the model is reliable, future data can be continuously ingested into the ML model and visualized through a real-time dashboard and alerting systems.

6.3. AI Expert System to Predict Application Performance

To further strengthen the machine learning models for performance test prediction, it can be equipped with an AI-based expert system that leverages Multiple-Criteria Decision Aid (MCDA) methodology; This methodology is applied to those evaluation problems where the final decision depends on many criteria [19]. The valuable data required for the expert system to enable decision-making is integrated with a software instrumentation agent that can thoroughly scan the entire application infrastructure resources and catalog all the critical configuration parameters that impact the system's Performance attributes, including the currently configured parameters and the maximum available capacity thresholds.[20]Some critical areas that impact performance are Server Hardware resources, CPU, Memory, Disc, JVM, DB Connection pool, etc.

These configuration values are transmitted to an AI-powered expert system, as depicted in Figure 5. AI-powered tools can run all this data through pre-set logic, deliver real-time insights into Application performance, and mitigate

risks associated with poor performance [21]. This system comprehensively analyzes the test result sets with various workload patterns and maps them to the resource utilization patterns to develop a Performance prediction model that can effectively predict current performance scalability & breaking points, capacity bottlenecks, and forecasting potential performance issues.

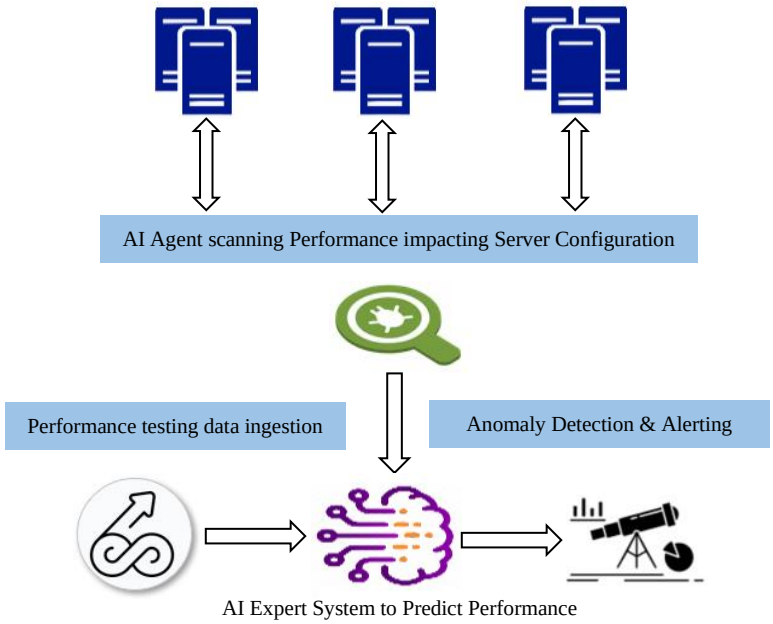

Fig. 5 Design of AI Expert System to Predict performance

The future of Performance Engineering entails the research and development of a lightweight, AI agent-based expert system that can predict and proactively address Performance issues in an end-to-end IT ecosystem. This innovative AI agent can monitor production traffic, design workload model, develop efficient test scripts, simulate production behavior in a test environment, identify the weakest link in the entire system, uncover performance bottlenecks, and provide valuable recommendations. While implementing such a cutting-edge AI system incurs significant research & development costs, the enormous demand for high-quality products that deliver exceptional performance and user experience will undoubtedly prompt organizations to make substantial investments in this space.

7. Conclusion

The research conducted in this paper highlights the challenges faced by performance testing in keeping pace with the rapid iterations of the Scaled Agile Framework (SAFe). It proposes solutions using DevOps automation, machine learning (ML), and artificial intelligence (AI). The research identifies the time-consuming phases in performance engineering and suggests opportunities for improvement; These include integrating performance testing tools into the CI/CD pipeline for faster test delivery, implementing operational intelligence for on-demand provisioning of performance environments, and automating the test data management process. Furthermore, this study emphasizes the transformative potential of AI and ML in performance testing to transcend traditional performance testing limitations and drive them into a new era of efficiency, accuracy, and innovation. Applying AI and ML in performance testing offers significant advantages, such as reducing script development time and enabling self-healing features. Moreover, the research underscores the significance of AI and ML in predicting system performance, as exemplified by the training dataset utilized in Amazon's cloud Machine Learning solution. Additionally, the paper presents a design for an AI-based expert system that can accurately predict software system performance using ML models and rule-based engines to forecast software system performance accurately, representing an exciting avenue for future research.

References

[1] Safe and Agile. [Online]. Available: https://scaledagile.com/what-is-safe/safe-and-agile/

[2] Rose de Fremery, How Dynatrace Empowers Performance Engineering Teams to Test at Scale, 2021. [Online]. Available: https://www.dynatrace.com/news/blog/how-dynatrace-empowers-performance-engineering-teams/

[3] Raluca Dovleac, "The Rise of DevQualOps and Implications on Software Quality," *International Journal of Computers*, vol. 8, pp. 5-10, 2023. [Google Scholar] [Publisher Link]
[4] Suresh Kannan Duraisamy, Bryce Bass, and Sai Mukkavilli, "Embedding Performance Testing in Agile Software Model," *International Journal of Software Engineering & Applications*, vol. 12, no. 6, pp. 1-11, 2021. [CrossRef] [Publisher Link]
[5] Pulasthi Perera, Roshali Silva, and Indika Perera, "Improve Software Quality Through Practicing DevOps," *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions*, 2018. [CrossRef] [Publisher Link]
[6] Sandesh Achar, "Enterprise SaaS Workloads on New-Generation Infrastructure-as-Code (IaC) on Multi-Cloud Platforms," *Global Disclosure of Economics and Business*, vol. 10, no. 2, pp. 55-74, 2021. [CrossRef] [Publisher Link]
[7] Praveen Bagare, and Desyatnikov, Test Data Management in Software Testing Life Cycle-Business Need and Benefits in Functional, Performance, and Automation Testing. [Google Scholar] [Publisher Link]
[8] Mantas Dvareckas, 5 Test Data Challenges that Every CTO should know about, 2022. [Online]. Available: https://www.curiositysoftware.ie/blog/5-test-data-challenges-every-cto-should-know-about
[9] Yogesh K. Dwivedi et al., "Artificial Intelligence (AI): Multidisciplinary Perspectives on Emerging Challenges, Opportunities, and Agenda for Research, Practice and Policy," *International Journal of Information Management*, vol. 57, p. 101994, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[10] S. Russell, and P. Norvig, "Artificial Intelligence: A Modern Approach. Pearson Education Limited," *Journal of Service Science and Management*, 2019.
[11] Yingnong Dang, Qingwei Lin, and Peng Huang, "AIOps: Real-World Challenges and Research Innovations," *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[12] Gururaj Hm, Power of AI in Performance Engineering, 2022. [Online]. Available: https://medium.com/@gururajhm/power-of-ai-in-performance-engineering-5ad56e0aa60b
[13] Samragyi Chamoli, Implementing AI for Improved Performance Testing, 2023. [Online]. Available: https://www.openxcell.com/blog/implementing-ai-for-improved-performance-testing/
[14] Dhaya Sindhu Battina, "Artificial Intelligence in Software Test Automation: A Systematic Literature Review," *International Journal of Emerging Technologies and Innovative Research*, vol. 6, no. 12, pp. 1329-1332, 2019. [Google Scholar] [Publisher Link]
[15] Perficient Latin America, How Machine Learning Enhances Performance Engineering and Testing, 2019. [Online]. Available: https://blogs.perficient.com/2019/11/21/how-machine-learning-enhances-performance-engineering-and-testing/
[16] Anita Maruti Patil-Nikam, Oriental Institute Usage of Machine Learning Algorithms in Software Testing, 2023. [Publisher Link]
[17] Lakshmisri Surya, "Machine Learning-future of Quality Assurance," *International Journal of Emerging Technologies and Innovative Research*, 2019. [Publisher Link]
[18] Garry Kranz, Amazon SageMaker. [Online]. Available: https://www.techtarget.com/searchaws/definition/Amazon-SageMaker
[19] I. Vlahavas et al., "ESSE: An Expert System for Software Evaluation," *Knowledge-based systems*, vol. 12, no. 4, pp. 183-197, 1999. [CrossRef] [Publisher Link]
[20] Fusion Middleware Performance and Tuning Guide. [Online]. Available: https://docs.oracle.com/cd/E52734_01/core/ASPER/top_issues.htm#ASPER99003
[21] Carmen Vitucci, and Draksha Sharma, Harnessing the Power of AI in Performance Engineering. [Online]. Available: https://www.qentelli.com/thought-leadership/insights/harnessing-power-ai-performance-engineering