SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V10I6P105

Volume 10 Issue 6, 36-42, June 2023
© 2023 Seventh Sense Research Group®

Original Article

Performance Testing using Machine Learning

Vivek Basavegowda Ramu

Independent Researcher, Connecticut, USA

Corresponding Author : vivekgowda.br@gmail.com

Received: 27 April 2023 Revised: 02 June 2023 Accepted: 14 June 2023 Published: 30 June 2023

Abstract *- Performance testing is a very important aspect of software development, aiming to ensure that applications meet the desired performance standards under various load conditions. Traditional performance testing approaches often face limitations and challenges in accurately identifying performance bottlenecks. This research investigates the idea of enhancing performance testing by utilizing machine learning techniques in order to go above these limits. This paper gives an overview of machine learning and some potential uses for it in performance evaluation. It discusses the benefits and advantages of incorporating machine learning, highlighting its ability to predict system behavior, detect anomalies and provide optimization recommendations. The paper also explores key performance metrics and data collection methods, emphasizing the significance of collecting accurate and relevant data for training machine learning models. The predictive modeling capabilities of machine learning are explored, showcasing how these models can be trained using historical performance data to forecast system behavior under different load scenarios. Techniques for evaluating the accuracy and effectiveness of predictive models are also discussed. The research also looks at the use of machine learning for performance anomaly detection, addressing the difficulties in locating performance-related issues. In order to identify and resolve performance bottlenecks, various methods, including outlier identification and grouping, are discussed. Additionally, the paper explores optimization and recommendation techniques driven by machine learning models. It highlights how these models can identify performance bottlenecks and provide suggestions for enhancing system performance, ultimately improving the user experience. By leveraging the capabilities of machine learning models, performance testers and software developers can enhance their ability to identify performance issues, optimize system performance and deliver efficient software.*

Keywords *- Performance testing, Machine learning, Predictive modeling, Anomaly detection, Optimization*.

1. Introduction

Performance testing plays a critical role in software development, ensuring that applications meet the desired performance standards and deliver optimal user experiences under various load conditions (Khan & Amjad, 2016). It involves evaluating the responsiveness, scalability, stability, and resource utilization of software systems (Basavegowda Ramu & Yeruva, 2023). However, traditional performance testing approaches face limitations and challenges in accurately identifying performance bottlenecks, leading to suboptimal system performance and potential customer dissatisfaction.

Traditional performance testing methods typically rely on predefined scenarios and thresholds (Costa et al., 2020), which may not effectively capture modern software applications' complex and dynamic nature. As a result, they may overlook hidden performance issues that emerge under real-world conditions. Additionally, manually analyzing performance test results can be time-consuming and subjective, making detecting and fixing performance issues challenging.

There has been an increase in interest in using machine learning approaches to improve performance testing to overcome these limits. Machine learning has emerged as a powerful technology that holds immense potential for revolutionizing performance testing in software development. As shown in Figure 1 (IBM Developer, 2020), It is classified as a subfield of artificial intelligence that focuses on developing algorithms capable of learning from data and making predictions or decisions without explicit programming (Duarte & Ståhl, 2018). By utilizing machine learning, performance testing procedures may be improved and modified to produce more precise and effective results. The capability of machine learning techniques to anticipate system behavior under diverse load levels is one of the major benefits of using them in performance assessment. Modern software systems are complicated and dynamic, yet traditional performance testing methods sometimes rely on predetermined scenarios and criteria, which may not adequately account for this. In contrast, machine learning models can analyze historical performance data and learn patterns and trends that enable them to make accurate predictions about system behavior. This predictive capability allows performance testers to proactively identify potential bottlenecks and optimize system performance before deploying the software.

Incorporating machine learning into performance testing processes has several objectives and goals (Moghadam et al., 2020). By utilizing data-driven methodologies, it primarily strives to increase the accuracy and efficacy of performance testing. Machine learning algorithms may find hidden insights and patterns in massive amounts of performance data that would be hard to spot manually. As a result, testers are more equipped to comprehend the variables affecting system performance and to make wiser choices. Additionally, performance testers can save time and effort by automating some performance testing processes using machine learning. Traditionally, analyzing performance test results and identifying performance issues has been a manual and time-consuming process. However, automated analysis and anomaly detection techniques can be employed with machine learning, enabling rapid identification of performance anomalies and potential bottlenecks. Another objective of incorporating machine learning is to optimize system performance by providing actionable recommendations. Machine learning models are able to identify performance bottlenecks and provide suitable optimization techniques based on the distinct characteristics of the application and its usage patterns. The overall user experience and system performance may be significantly enhanced as a result.

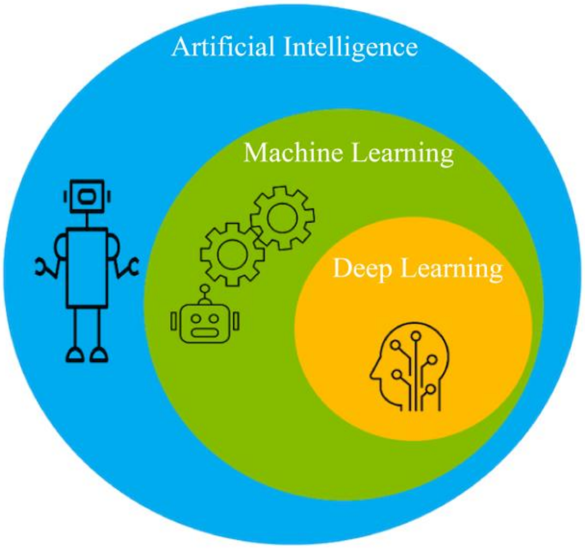

Fig. 1 Machine learning is a subfield of Artificial intelligence

2. Literature Review

The significance of fulfilling performance criteria for performance-critical software systems was highlighted by (Helali Moghadam et al., 2019). Although model-based analysis is frequently employed, it can be difficult to derive a correct performance model, especially for complex systems. The study suggests a paradigm for a performance assessment based on self-adaptive learning as an alternative. In order to determine the performance breaking point for different software systems, the framework learns how to do stress testing. It discovers the best strategy for producing stress test cases and then repeats it to produce test cases more quickly. According to the study, the suggested framework is adaptable to various software system types, making it easier to conduct autonomous performance testing. However, the suggested learning-based paradigm is not thoroughly assessed or validated in the research, which may restrict its general application and efficacy.

In a different research (Moghadam, 2019) suggests an autonomous self-adaptive stress testing framework that makes use of model-free reinforcement learning to discover the best strategy for generating stress test cases independent of system models. The framework attempts to increase stress testing's effectiveness and flexibility, especially for complex software systems. Experimental research shows how well the suggested framework performs without using performance models in creating stress test scenarios for different software systems. However, the research does not include a thorough analysis of the framework's performance or address any potential issues in real-world situations, which might restrict its usefulness.

ACTA, an automated test-generating technique for black-box performance testing of substantial and intricate software systems, was introduced by (Sedaghatbaf et al., 2021). Even in the absence of a sizable body of previous test data, ACTA uses active learning and uncertainty sampling to choose which tests to run dynamically. To produce tests based on predetermined performance requirements, the approach uses conditional generative adversarial networks. Experimental testing on a benchmark web application reveals that ACTA outperforms random testing, PerfXRL, and DN, two further machine learning techniques. However, the study does not provide an extensive comparison with other existing methods or address potential limitations or challenges in applying ACTA to real-world scenarios, which could limit its generalizability and practical usefulness.

A study (Kumbakonam & Shafi, 2021) discusses the importance of performance measurement and software quality testing in developing high-standard applications. It highlights the use of machine learning for effective testing and introduces image processing techniques to minimize waiting time. However, the study lacks specific details on implementing and evaluating these techniques, limiting its ability to provide concrete insights and evidence of their effectiveness in developing a "perfect" application.

3. Methodology

The section provides a comprehensive understanding of how machine learning techniques can be employed in performance testing, covering aspects such as data collection, predictive modeling, anomaly detection and Optimization. By adopting these methodologies, performance testers can harness the power of machine learning to enhance their ability to detect performance issues, optimize system performance and deliver efficient software.

3.1. Key Performance Metrics and Data Collection

In order to derive actionable insights from performance data in performance testing, it is essential to use the right machine learning algorithms. For better comprehension and software system optimization, several machine-learning techniques may be used to monitor and interpret performance information. Let's explore some common machine learning algorithms and their relevance in performance testing.

- Supervised learning is a popular category of machine learning algorithms ("Beyond Supervised Learning," 2021) and has wider adoption in performance testing. It involves training models using labeled data, where the desired output or performance metric is known. These models learn patterns and relationships from historical performance data and can make predictions based on new input. For example, regression algorithms such as linear regression and support vector regression can be utilized to predict performance metrics like response time or throughput based on various input parameters.

- Unsupervised Learning algorithms, on the other hand, do not rely on labeled data (Celebi & Aydin, 2016). They are designed to identify hidden structures and patterns within performance data. The type of clustering algorithms, such as k-means clustering or hierarchical clustering, has the ability to group similar performance data points together, enabling performance testers to identify distinct performance profiles or patterns in the system behavior. This can help in identifying outliers or anomalies that may require further investigation.

- Reinforcement Learning is a subset of machine learning (Gottesman et al., 2019), which can also be employed in performance testing. It involves training models which can make sequential decisions based on the feedback from the environment. In the context of performance testing, reinforcement learning algorithms can optimize system performance by taking actions that maximize a specified performance metric. These algorithms learn through trial and error and adapt their decision-making based on the received rewards or penalties.

Historical performance data is crucial to train these machine learning models. Performance testers collect data during various load tests (Lenka et al., 2018), capturing metrics such as response time, throughput, CPU utilization, memory usage, and network latency. This historical data acts as a training set for the machine learning algorithms. The models learn from this data, identifying patterns and correlations between input parameters and performance metrics. This training phase enables the models to make accurate predictions or detect anomalies in real-time scenarios. Data preprocessing is an essential step before training the models. It involves cleaning the data, handling missing values, normalizing the values, and selecting relevant features for training. Once the data is prepared, the models are trained using various optimization techniques and algorithms. The training process involves adjusting model parameters to minimize errors and optimize the performance of the models. Figure 2 shows the graphical representation of unsupervised, supervised and reinforcement learning (Arya, 2022).

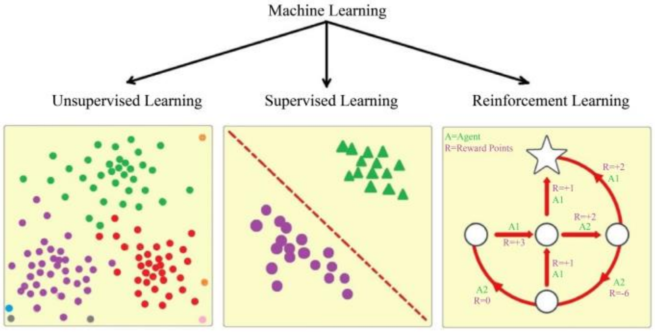

Fig. 2 Unsupervised, Supervised and Reinforcement learning

3.2. Predictive Modeling for Performance Testing

Predictive modeling is the process of creating statistical models to estimate the likelihood and trends of future occurrences (Zhu et al., 2016). Machine learning models can forecast system behavior under various load circumstances, giving useful information about how software applications will function in real-world conditions. By training these models on historical performance data, performance testers can obtain a very deep understanding of system performance which will help them to make informed decisions to optimize software applications. Let's delve into the details of predictive modeling for performance testing. Machine learning models leverage historical performance data, which will help them to learn patterns and the relationships between input parameters and performance metrics. Models can forecast how the system will operate under various load circumstances by examining this data. For instance, the model may predict performance measures like response time, throughput, or resource consumption by inputting the number of concurrent users, transaction volume, or network bandwidth. This predictive capability helps performance testers proactively identify potential performance issues and take appropriate measures to optimize system performance.

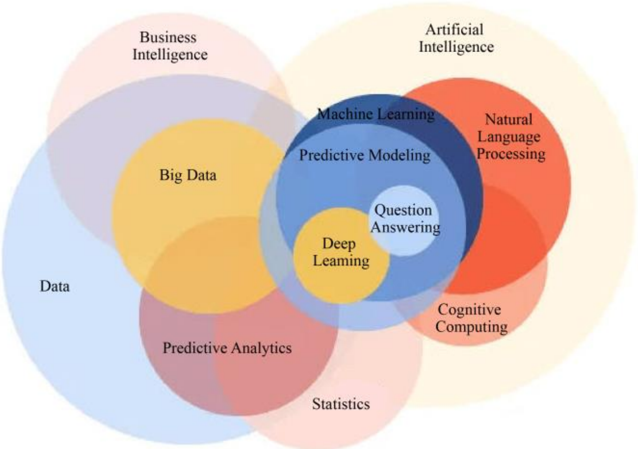

Fig. 3 Predictive modeling position among data and machine learning

Several factors and features are considered to build accurate and reliable predictive models in the modelling process. Firstly, selecting relevant performance metrics to predict is crucial. These metrics can vary depending on the specific software application but commonly include response time, throughput, latency, and error rates. Additionally, factors such as hardware specifications, network configurations, and software configurations are considered input features for the predictive models. Performance testers must carefully identify and collect these features to ensure the models capture the relevant factors influencing system performance. Performance testing must include evaluating prediction models for accuracy and efficiency. The effectiveness of the models may be evaluated using a variety of ways. Cross-validation is frequently employed to evaluate the model's generalization by dividing the dataset into training and testing sets. The evaluation of the testing set and model is trained on the training set to measure its performance against the unseen data. To measure the model's precision and predictive power, metrics like the coefficient of determination (R-squared), mean squared error, or mean absolute error can be utilized. Additionally, methods like holdout validation and k-fold cross-validation can offer more information on the resilience and dependability of the model. Figure 3 (What Is Predictive Modeling? Definition and Overview, 2023) showcases how predictive modeling fits in the world of data and machine learning.

3.3. Performance Anomaly Detection

Modern information systems are hampered by a fundamental issue called performance anomalies, which has an impact on how hosted applications run (Baril et al., 2020). Machine learning techniques offer a powerful approach to detecting performance anomalies and identifying bottlenecks in software applications. By leveraging these techniques, performance testers can proactively identify deviations from normal behavior and address potential performance issues. Let's delve into how machine learning can be utilized for performance anomaly detection, the challenges involved, and the different approaches available. Machine learning techniques excel in detecting performance anomalies by leveraging historical performance data. These techniques analyze patterns, trends, and relationships within the data to identify instances where the system deviates from expected behavior. By training machine learning models on historical performance data, they can learn the normal performance patterns and use that knowledge to detect anomalies in real-time scenarios. This allows performance testers to address performance issues and optimize system performance promptly.

However, performance anomaly detection comes with its own set of challenges. One major challenge is the variability and complexity of performance data. Performance metrics can exhibit intricate relationships influenced by factors such as workload patterns, system configurations, and user behavior. Identifying anomalies accurately requires machine learning models to capture these intricate relationships and distinguish between normal and abnormal behavior. Another challenge is the availability and quality of labeled data. Obtaining labeled data that accurately represent performance anomalies can be difficult and time-consuming. Performance testers often need to rely on expert knowledge and domain expertise to label data, which can introduce subjectivity and potential inaccuracies.

Different approaches can be employed in performance anomaly detection to overcome these challenges. Outlier detection is one approach where machine learning models identify data points that deviate significantly from normal behavior. These outliers represent potential performance anomalies and can help pinpoint performance issues. Clustering techniques can also be utilized to group similar performance data points together. Performance testers can identify abnormal behavior patterns and potential bottlenecks within specific groups by examining the clusters.

Additionally, time series analysis is an effective approach for performance anomaly detection. This technique considers the sequential nature of performance data, capturing temporal dependencies and trends. Machine learning models can identify deviations from expected performance behavior by analyzing the patterns and fluctuations over time. This approach is particularly useful when detecting performance anomalies in dynamic systems where performance patterns can change over time.

3.4. Optimization and Recommendation Techniques

Performance optimization can often pose a real challenge (Minami, 2019). Machine learning models have the potential to provide valuable performance optimization recommendations, assisting performance testers in identifying and addressing performance bottlenecks. By leveraging the capabilities of these models, performance testers can enhance system performance and deliver efficient software. Let's explore how machine learning can provide optimization recommendations, explore ways to identify performance bottlenecks, and highlight the benefits of using machine learning optimization. Machine learning algorithms can analyze operational data to identify areas of operational bottlenecks. These models learn patterns and correlations between performance metrics and various factors such as system configurations, workload characteristics, or resource utilization. By examining these relationships, machine learning models can pinpoint the specific factors that contribute to performance bottlenecks. For example, a model might identify that high CPU utilization or inefficient database queries are causing performance degradation.

Based on these insights, machine learning models can suggest possible solutions for optimizing system performance. For instance, the model might recommend optimizing database queries, increasing hardware resources, or fine-tuning system configurations. These recommendations are derived from the patterns and relationships identified by the model, offering specific guidance on areas that can be improved to enhance performance. Identifying performance bottlenecks and suggesting solutions involve various techniques. Machine learning models can utilize feature importance analysis to determine the most influential factors contributing to performance degradation. By understanding the relative importance of different features, performance testers can prioritize optimization efforts. Furthermore, sensitivity analysis can be performed to evaluate how changes in certain factors impact system performance. This helps identify critical areas that require attention and guides the optimization process.

Machine learning-driven optimization techniques offer several benefits. Firstly, they enable performance testers to leverage historical performance data to gain insights and make data-driven decisions. By learning from the data, machine learning models can provide targeted recommendations tailored to the specific characteristics of the software application. This leads to more effective and efficient optimization efforts, saving time and resources. Additionally, machine learning models can simultaneously handle the complexity and variability of performance data, considering multiple factors. They can capture intricate relationships and interactions between different parameters, providing a holistic view of system performance. This comprehensive understanding allows for more accurate identification of bottlenecks and effective optimization strategies.

Moreover, machine learning-driven optimization techniques can adapt and evolve over time. As new performance data becomes available, the models can be retrained to incorporate the latest insights and trends. This adaptability ensures that optimization recommendations remain up-to-date and relevant, accommodating changes in workload patterns, user behavior, and system configurations.

4. Results

The incorporation of techniques in machine learning for performance testing has demonstrated promising outcomes in enhancing the accuracy and efficiency of performance analysis and Optimization. By leveraging historical performance data, machine learning models offer valuable insights, anomaly detection, and optimization recommendations to improve system performance. Various machine learning methods can be used to assess performance indicators and derive valuable insights, including supervised learning, unsupervised learning and reinforcement learning. Based on different input characteristics, supervised learning algorithms, including regression techniques like support vector and linear regression, can forecast performance metrics like response time and throughput. Unsupervised learning algorithms may find patterns and group comparable performance data points together to help in anomaly detection. Examples of these algorithms are hierarchical and k-means clustering. By making a series of judgments based on input from the environment, reinforcement learning algorithms can improve system performance. Historical performance data, encompassing metrics like response time, throughput, CPU utilization, memory usage, and network latency, is collected during load tests to train machine learning models. Preprocessing steps like data cleaning, handling missing values, normalization, and feature selection are performed to prepare the data for training the models.

The benefits of machine learning-driven optimization techniques include data-driven decision-making, handling complexity and variability in performance data, and adaptability over time. By leveraging historical performance data, machine learning models offer targeted recommendations tailored to the specific characteristics of the software application, resulting in more effective optimization efforts. These models can capture intricate relationships and interactions between parameters, providing a comprehensive view of system performance. Furthermore, the adaptability of machine learning models ensures that optimization recommendations remain up-to-date and relevant as new performance data becomes available.

5. Limitations and Future Study

Utilizing machine learning in performance testing presents both challenges and limitations that need to be addressed. By understanding these challenges, identifying open research questions and encouraging future exploration, we can further advance the field. One of the major challenges of utilizing machine learning in performance testing is the availability of high-quality and labeled training data. Building accurate and reliable machine learning models requires extensive and diverse performance data. However, acquiring such data can be challenging, especially when considering the need for labeled data representing various performance scenarios and anomalies. Performance testers often need to rely on expert knowledge and domain expertise to label data, introducing subjectivity and potential inaccuracies. Addressing this challenge requires the development of standardized datasets and collaboration among researchers and industry practitioners to ensure the availability of quality training data.

Another challenge lies in the interpretability of machine learning models. Performance testers and stakeholders often require explanations for these models' predictions and recommendations. Having trust and confidence in the results requires an understanding of the underlying variables and connections that influence the model's judgments. Further research is needed in the crucial field of improving the machine learning models' interpretability in the context of performance assessment. Techniques such as rule extraction, feature importance analysis and model visualization can help shed light on the decision-making process of these models.

Furthermore, it continues to be difficult to generalize machine learning models across various software systems and applications. It might be challenging to develop models that can adapt to various circumstances due to software systems' wide variation in performance characteristics. Further research is needed to explore transfer learning and domain adaptation techniques that enable models to leverage knowledge gained from one application to improve performance testing in other similar applications.

Identifying open research questions is vital for advancing the field of machine learning in performance testing. For example, investigating the impact of different feature selection and feature engineering techniques on model performance can help optimize the modeling process. Exploring ensemble learning approaches, which combine multiple models, can enhance the robustness and accuracy of performance predictions. Moreover, investigating the scalability and efficiency of machine learning models for large-scale performance testing scenarios is crucial, as real-world systems often handle massive user loads and complex architectures. Collaboration between academia and industry is essential to encourage future research and exploration. Collaborative efforts can help bridge the gap between theoretical advancements and practical implementations. Researchers can work closely with performance testers and software developers to understand real-world challenges and develop solutions applicable in industry settings. Moreover, sharing datasets, benchmarks, and evaluation metrics will facilitate the comparison and replication of studies, fostering a collective effort to advance the field of machine learning in performance testing.

6. Conclusion

The tremendous potential and significance of incorporating machine learning approaches into performance assessment have been highlighted by this study. Machine learning models are essential in improving the accuracy and efficiency of performance analysis and Optimization by utilizing past performance data. The main conclusions drawn from this study emphasize the many uses of machine learning algorithms, like supervised learning, unsupervised learning and reinforcement learning, in examining performance measures and deriving insightful conclusions. These models have proven to be efficient in forecasting performance measures, spotting abnormalities, and making optimization suggestions specifically catered to software applications' specific features.

The findings of this research underscore the transformative impact of machine learning in performance testing. By leveraging advanced algorithms and computational power, performance testers can make data-driven decisions and optimize system performance more effectively. The ability to accurately forecast system behavior under different load scenarios through predictive modeling opens up new avenues for proactive performance management. Additionally, machine learning algorithms are excellent at spotting performance anomalies, enabling quick identification and potential bottleneck mitigation. It is needed to identify the value of continued research and development in this specific area. The importance of machine learning in performance testing increases as technology advances and software systems become more complex. The capabilities of machine learning models in this field will continue to be further improved by ongoing research into innovative algorithms, data pretreatment methods, and optimization approaches. Collaboration between researchers, practitioners, and industry experts is vital to drive innovation and ensure the successful adoption of machine learning-driven performance testing practices. Integrating machine learning in performance testing offers a transformative approach to optimizing system performance. The conclusions and findings presented in this research showcase the tremendous potential of machine learning in this field. As we move forward, continued efforts in research and development will push this industry to new heights, helping organizations achieve higher levels of productivity, efficiency and reliability in their software applications. This paper proposes the solid utilization of machine learning techniques in performance testing, enabling accurate prediction of system behavior, detection of anomalies and provision of optimization recommendations, surpassing the limitations of traditional approaches mentioned in the literature review.

References

[1] Rijwan Khan, and Mohd Amjad, "Performance Testing (Load) of Web Applications Based on Test Case Management," *Perspectives in Science*, vol. 8, pp. 355–357, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Vivek Basavegowda Ramu, and Ajay Reddy Yeruva, "The Capability of Observing Performance in Healthcare Systems," *Computational Intelligence for Clinical Diagnosis, EAI/Springer Innovations in Communication and Computing*, Springer, Cham, pp. 541-548, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Victor Costa, "Taxonomy of Performance Testing Tools," *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 1997-2004, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] IBM Developer, IBM Developer, 2020. [Online]. Available: Https://Developer.Ibm.Com/Learningpaths/Get-Started-With-Deep-Learning/An-Introduction-to-Deep-Learning/

[5] Denio Duarte, and Niclas Ståhl, "Machine Learning: A Concise Overview," *Studies in Big Data*, pp. 27–58. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mahshid Helali Moghadam et al., "Poster: Performance Testing Driven By Reinforcement Learning," *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, pp. 402-405, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mahshid Helali Moghadam et al., "Machine Learning to Guide Performance Testing: An Autonomous Test Framework," *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 164-167, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mahshid Helali Moghadam, "Machine Learning-Assisted Performance Testing," *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1187-1189, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ali Sedaghatbaf et al., "Automated Performance Testing Based on Active Deep Learning," *2021 IEEE/ACM International Conference on Automation of Software Test (AST*), pp. 11-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Munidhanalakshmi Kumbakonam, and R. Mahammad Shafi, "Application Measurement and Software Quality Testing Using Machine Learning Performance Techniques," *2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, pp. 372-376, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] *Beyond Supervised Learning*, Deep Learning for Physics Research, pp. 217–218, 2021. [CrossRef] [Publisher Link]

[12] M. Emre Celebi, and Kemal Aydin, *Unsupervised Learning Algorithms*, 2016. [CrossRef] [Publisher Link]

[13] Omer Gottesman et al., "Guidelines for Reinforcement Learning in Healthcare," *Nature Medicine*, vol. 25, pp. 16–18, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Rakesh Kumar Lenka et al., "Performance and Load Testing: Tools and Challenges," *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, pp. 2257-2261, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Nisha Arya, A Brief Introduction to Reinforcement Learning. Ejable, 2022. [Online]. Available: Https://Www.Ejable.Com/Tech-Corner/Ai-Machine-Learning-and-Deep-Learning/A-Brief-Introduction-to-Reinforcement-Learning/

[16] Qiang Zhu, "Business Applications of Predictive Modeling at Scale," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2139–2140, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[17] What Is Predictive Modeling? Definition and Overview, What Is Predictive Modeling? Definition and Overview | Outsystems, 2023. [Online]. Avaialable: Https://Www.Outsystems.Com/Glossary/What-Is-Predictive-Modeling/

[18] Xavier Baril et al., "Application Performance Anomaly Detection with LSTM on Temporal Irregularities in Logs," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1961-1964, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Kazuo Minami et al., "Performance Optimization of Applications," *The Art of High Performance Computing for Computational Science,* vol. 2, pp. 11–39, 2019. [CrossRef] [Publisher Link]