INDIAN STREAMS RESEARCH JOURNAL
ISSN NO : 2230-7850
IMPACT FACTOR : 5.1651 (UIF)
VOLUME - 13 | ISSUE - 6 | JULY - 2023

AI-DRIVEN ANALYSIS OF SOFTWARE DEMONSTRATIONS FOR ARCHITECTURAL DECISION-MAKING

Amaresh S/O Ghalappa
Research Scholar

Dr. Shashi
Guide
Professor, Chaudhary Charansing University Meerut.

ABSTRACT:

Decisions about software architecture have a big impact on the scalability, maintainability, and performance of systems. Conventional decision-making procedures depend on human assessment of software demos and expert judgment, both of which can be laborious and biased. In order to assist with architectural decision-making, this paper investigates an AI-driven method of software demonstration analysis. Our framework analyzes architectural trade-offs, extracts important insights from software presentations, and makes data-driven recommendations by utilizing machine learning and natural language processing techniques. This approach speeds up the evaluation process, lowers subjectivity, and improves decision accuracy, making it a useful tool for development teams and architects.

KEYWORDS : *Natural language processing (NLP), software architecture, machine learning (ML), artificial intelligence (AI), and architectural decision-making.*

INTRODUCTION

Because it affects a system's scalability, maintainability, and performance, software architecture is crucial to its success. Expert judgment, manual evaluation, and qualitative evaluations of software demonstrations are all major components of traditional architectural decision-making. These techniques, however, can be laborious, erratic, and biased by people. The swift development of artificial intelligence (AI) offers a chance to improve this procedure by automating software demonstration analysis and offering insights based on data. In order to extract pertinent information from software demonstrations, AI-driven analysis makes use of machine learning (ML), natural language processing (NLP), and pattern recognition. This method makes it possible to identify important architectural elements, performance indicators, and trade-offs that influence choices. AI can recognize patterns, correlate results with best practices, and recommend the best architectural options by analyzing both textual and visual data. By incorporating AI into architectural decision-making, evaluations become more accurate and less subjective. Instead of depending only on gut feeling or prior experiences, it enables development teams and architects to make well-informed decisions based on empirical data. AI-powered systems can also scale effectively, analyzing vast amounts of software demos to find patterns and insights across several projects.

AIMS AND OBJECTIVES

The main goal of this study is to investigate how automating the assessment of software demonstrations through AI-driven analysis can improve software architectural decision-making. The goal of this project is to create a framework that uses artificial intelligence—such as machine learning and natural language processing—to help architects make data-driven decisions by gleaning valuable insights from software presentations. In order to provide unbiased and scalable decision support, this research focuses on identifying important architectural elements, trade-offs, and performance metrics from software demonstrations. By incorporating AI-driven approaches, it seeks to lessen human biases, increase accuracy, and speed up the decision-making process.

The study also aims to determine how well AI can evaluate various architectural styles, spot trends in numerous software demonstrations, and correlate results with industry best practices. By doing this, the study hopes to develop an automated, structured method that improves on conventional software architecture assessment techniques. The study also intends to address issues like data quality, result interpretability, and integration with current decision-making workflows that are associated with AI-driven analysis. The ultimate objective is to advance software architecture by offering a cutting-edge, AI-powered solution that increases the precision and efficiency of decision-making.

LITERATURE REVIEW

More effective and data-driven methods of architectural decision-making are required due to the growing complexity of contemporary software systems. Conventional approaches frequently suffer from subjectivity, inconsistencies, and time inefficiencies because they mainly rely on expert judgment and manual evaluation of software demonstrations. Recent developments in artificial intelligence (AI), specifically in the areas of natural language processing (NLP) and machine learning (ML), have opened up new avenues for automating software demonstration analysis to aid in architectural decision-making. Current studies on AI-powered decision support systems demonstrate how well AI can extract and analyze vast amounts of architectural data. Research on NLP applications in software engineering shows how AI can analyze documentation, code snippets, and textual descriptions to find important architectural patterns. The application of machine learning (ML) techniques to automated architectural reviews, anomaly detection, and software performance evaluation has also shown encouraging results in terms of increasing decision accuracy and decreasing human labor.

Numerous studies concentrate on AI-powered tools that identify key performance indicators, scalability issues, and maintainability metrics to aid in architectural trade-off analysis. This field of study focuses on how AI can identify trends in previous software architectures and link them to effective implementations. Additionally investigated are AI-based software architecture recommendation systems that use best practices and historical data to inform choices. Even with these developments, incorporating AI into architectural decision-making processes still presents difficulties. Significant obstacles include the need for domain-specific training models, problems with data quality, and the interpretability of insights produced by AI. Furthermore, even though AI can increase productivity, its suggestions need to be thoroughly verified against contextual requirements to guarantee dependability.

RESEARCH METHOLOGY

In order to investigate the use of AI-driven analysis in software architectural decision-making, this study employs a methodical approach. It analyzes software demonstrations and derives valuable insights for decision support by combining data collection, machine learning methods, and natural language processing (NLP). Data collection from multiple sources, such as technical documentation, architectural presentations, and recorded software demonstrations, is the first step in the study. To eliminate noise and standardize formats for analysis, these datasets undergo preprocessing. Relevant textual information, such as architectural patterns, key performance indicators, and trade-offs, is extracted using natural language processing (NLP) techniques. In order to find patterns and connections between various architectural choices and system results, machine learning models are simultaneously trained on both structured and unstructured data.

Based on past data, a supervised learning technique is applied to categorize architectural elements and forecast decision outcomes. Additionally, to find hidden patterns in software demonstrations, unsupervised learning techniques like anomaly detection and clustering are used. The analysis framework evaluates qualitative elements of software presentations, including stakeholder concerns and developer intent, by combining topic modeling and AI-driven sentiment analysis. Through case studies and benchmark comparisons, the research carries out empirical evaluations to verify the efficacy of AI-driven analysis. The accuracy and dependability of the suggested AI models are evaluated using real-world software demonstrations, and their suggestions are contrasted with professional evaluations. The models are assessed using performance metrics such as computational efficiency, precision, and recall. To guarantee impartial and open decision-making, ethical factors are taken into account, such as data privacy and bias reduction. The study also looks at how AI-driven insights can be incorporated into current architectural workflows, showing how these techniques can enhance rather than replace human expertise. This approach seeks to create a solid, automated framework for evaluating software demos by utilizing AI techniques, which will ultimately increase the precision and efficiency of architectural decision-making procedures.

STATEMENT OF THE PROBLEM

Making decisions about software architecture is a complicated process that calls for careful consideration of a number of variables, such as system security, scalability, maintainability, and performance. These choices have historically been made using expert judgment, manual software demonstration analysis, and qualitative evaluations—all of which can be laborious, erratic, and biased. More data-driven, scalable, and objective methods of making architectural decisions are required as software systems become more complex. Software demonstration analysis techniques currently in use frequently lack automation and do not take advantage of the enormous volumes of data available in contemporary software engineering. This restriction leads to less accurate architectural evaluations, ineffective decision-making procedures, and a higher chance of making less-than-ideal design decisions. Moreover, incomplete assessments result from existing evaluation frameworks' inability to handle unstructured data from software demonstrations, including technical presentations, code walkthroughs, and architectural discussions.

New opportunities to improve software architecture analysis are presented by developments in artificial intelligence (AI), specifically in machine learning (ML) and natural language processing (NLP). AI-driven analysis can find architectural patterns, evaluate trade-offs in a methodical and data-driven way, and automatically extract insights from software demonstrations. The use of AI in this field is still in its infancy, though, and there are still issues with interpretability, dependability, and integration with current decision-making processes. The goal of this research is to create an AI-driven framework for software demonstration analysis in order to overcome the drawbacks of conventional architectural decision-making. It seeks to lessen human biases while improving architectural evaluations' precision, effectiveness, and consistency. This study investigates how automated analysis can help software architects make more unbiased, scalable, and well-informed decisions by utilizing AI techniques.

DISCUSSION

An important change from conventional, experience-driven assessments to data-driven, automated analysis is represented by the incorporation of artificial intelligence (AI) into software architecture decision-making. A more methodical and scalable way to evaluate architectural trade-offs, performance implications, and best practices is provided by AI-driven approaches that use machine learning (ML) and natural language processing (NLP) to extract insights from software demonstrations. The capacity of AI-driven analysis to handle enormous volumes of both structured and unstructured data from software demonstrations is one of its main advantages. AI is able to recognize patterns, spot anomalies, and link architectural choices to system results, in contrast to conventional approaches that depend on human interpretation. This improves the accuracy of architectural evaluations and lessens subjectivity. Furthermore, comprehensive decision support is made possible by AI-powered tools that can analyze technical discussions, presentation transcripts, and software documentation.

Nevertheless, there are obstacles to using AI in architectural decision-making. Because architects need to trust and comprehend the recommendations made by machine learning models, the interpretability of AI-generated insights is still a concern. Furthermore, biases in data can produce skewed results, and AI systems need high-quality training data. To overcome these obstacles, explainable AI models and strong validation methods are needed to make sure AI-driven suggestions adhere to best practices and industry standards. The incorporation of AI-driven analysis into current software development processes is another crucial factor to take into account. AI should supplement human expertise rather than replace it, even though it can improve decision-making. A hybrid strategy can achieve a balance between automation and human judgment by having AI offer data-driven insights and architects base their final decisions on contextual considerations. This conversation demonstrates how AI-driven analysis has the potential to transform software architecture decision-making by increasing productivity, decreasing human error, and facilitating better decision-making. AI is a useful tool in the field of software architecture, despite current obstacles that can be addressed with improved integration techniques and ongoing developments in AI methodologies.

CONCLUSION

AI-powered software demonstration analysis offers a revolutionary method of architectural decision-making, replacing manual, conventional assessments with automated, data-driven insights. AI improves the precision, effectiveness, and scalability of architectural evaluations by utilizing machine learning and natural language processing, empowering software architects to make more unbiased and knowledgeable choices. The study demonstrates how AI can be used to evaluate trade-offs, find architectural patterns, and glean valuable insights from software demonstrations. AI-driven approaches offer a more methodical framework for evaluation, minimize human biases, and expedite decision-making. To guarantee trustworthy and useful recommendations, however, issues like interpretability, data quality, and integration with current workflows must be resolved.

Despite these obstacles, AI has the potential to support architects with empirical data and predictive analysis, enhancing rather than replacing human expertise. AI's role in software architecture evaluation will be further strengthened by future developments in explainability, better training datasets, and improved integration mechanisms.All things considered, AI-driven analysis is a major breakthrough in architectural decision-making, offering a methodical and scalable approach that enhances software quality, maximizes design options, and speeds up the assessment procedure. AI will become an essential tool for contemporary software development as its use in software architecture improves with further development.

REFERENCES

1. Bass, L., Clements, P., &Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley.
2. Garlan, D., & Shaw, M. (1994). An Introduction to Software Architecture.Carnegie Mellon University.
3. Chen, L., Babar, M. A., & Ali, N. (2019). "Architectural Decision-Making: Current and Emerging Research Directions,"
4. Guo, J., Ali, N., & Liang, P. (2020). "AI-assisted Software Engineering: Trends and Challenges,"
5. Zimmermann, O., Wegmann, P., &Koziolek, H. (2021). "Architectural Decision Guidance Across Projects:
6. Hassan, S., & Shang, W. (2020). "Machine Learning for Software Architecture Analysis: A Systematic Mapping Study,"