LONDON JOURNAL OF INTERDISCIPLINARY SCIENCES

eISSN 3049-7809

Examination of AI Enhanced Distributed Systems and its Effects on Software Engineering

Mehmet Uzgoren*
Hana Sultana**
Mina Uzgoren***
Nilufer Okumus****

Abstract
Distributed systems are groups of independent computers that appear to the system's users as a single coherent system. These systems involve several critical factors, including network communication, concurrency, and fault tolerance, which are vital in software engineering. For example, cloud computing platforms like Amazon Web Services (AWS) use distributed systems to offer scalable, on-demand resources to millions of users worldwide. The introduction of Artificial Intelligence (AI) into software engineering marks a transformative period that reshapes traditional development processes and breaks down the complexity of distributed systems. AI automates routine tasks and simplifies complex processes, serving as a digital collaborator that enables developers to focus on strategic thinking and creativity. One significant advantage of using AI in distributed systems within software engineering is enhanced resource optimization. AI can identify irregularities that might indicate hardware malfunctions and take preventative measures to address them, such as rerouting traffic to healthier servers. However, distributed systems also have challenges, including increased complexity in system design, difficulty ensuring data consistency, and potential security vulnerabilities. The intricate structure of these systems can result in problems with high startup costs, safety risks, and the accuracy of the data provided. Integrating AI into distributed systems offers both significant advantages and disadvantages. This study evaluates how AI impacts the efficiency and security of distributed systems in software engineering. By analyzing the advantages and disadvantages of AI-enhanced distributed systems, we can gain a comprehensive understanding of their overall effectiveness and implications in the field.
Keywords: Artificial Intelligence (AI), Distributed Systems, Software Engineering

https://doi.org/10.31039/ljis.2024.3.295

*University of Texas, Austin, USA
**HS of Discovery, USA, hanasu929@gmail.com
***HS of Innovation, Sugar Land, USA

Introduction

Artificial intelligence (AI) encompasses a wide range of technologies and approaches aimed at performing tasks that require human intelligence. The transformative impact of AI extends beyond software engineering into diverse sectors, revolutionizing industries by enhancing efficiency and productivity. As AI algorithms advance, their ability to understand context, learn coding patterns, and generate complex sections of code has improved. This progress has raised concerns among software developers regarding the potential of AI to replace their coding tasks. A key driver of AI advancements is the development of distributed systems. These systems improve the overall performance of AI by enabling faster processing and more efficient implementation of machine learning models. Distributed systems are important when a single computer cannot handle the workload, providing features that are difficult or impossible to achieve with a single system. They enable parallel processing, fault tolerance, and scalability, which are essential for managing large-scale AI applications. The integration of AI into distributed systems within software engineering leads to the development of more robust, intelligent, and efficient systems that can manage the complexities and requirements of modern applications. Distributed systems enable AI to process vast amounts of data, perform complex computations rapidly, and optimize resource usage. This synergy enhances the capabilities of AI, making it more efficient and versatile. This paper explores the evolving roles of AI in the workplace, focusing on its integration into distributed systems within software engineering. It investigates how distributed systems are driving advancements in AI and examines the broader impact these developments will have on the field of software engineering as a whole.

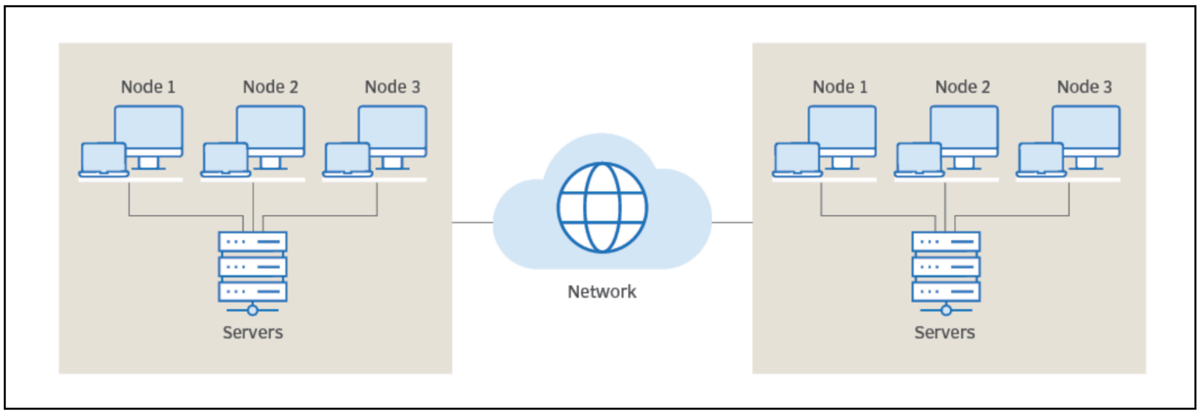

Figure 1. Illustration of the distributed computing process within servers and network (Yasar, 2024)

Problem Statement

While AI in distributed systems offers significant benefits to software engineering, its adoption also presents several challenges that must be addressed. The complexity of development and maintenance can increase by 70% due to AI. Ensuring sustainable integration requires addressing critical issues such as data privacy, security risks, significant resource demands, and ethical concerns, which are essential for equitable advancements in the field.

Background Research

The history of integrating AI with distributed systems in software engineering spans several decades. In the early days of AI research (1950s and 1960s), computing power was limited, and AI algorithms were primarily executed on single machines or small networks. As AI techniques like expert systems and neural networks gained traction in the 1970s and 1980s, researchers began exploring distributed computing architectures.

By the 1990s, advances in networking technologies and the emergence of the internet spurred further interest in distributed systems for AI applications. This era saw the development of distributed AI systems for tasks such as distributed reasoning and collaborative filtering. The early 2000s marked a significant turning point with grid computing and the rise of cloud computing, which democratized access to scalable computing resources, enabling more complex AI models on distributed architectures. Distributed systems operate by sharing and replicating data through continuous information streams connecting different nodes, thereby ensuring data consistency and accuracy. This approach improves performance compared to a single computing system processing all data itself. Distributed computing allows multiple nodes to work on separate tasks, enabling more efficient processing and task distribution (Gillis 2024). Distributed systems are used in networks, servers, mobile applications, and many other internet processes, storing large amounts of data for quick and efficient access. Despite having multiple nodes, distributed systems work as a unified system, processing data as one unit. This capability has enabled many technological innovations, particularly in AI, which leverages distributed systems to collect and distribute vast amounts of information quickly (Mwase, 2024).

Impact on Software Engineering

Distributed systems have revolutionized software engineering by introducing architectural complexity and necessitating new design patterns. They offer improved scalability and performance, essential for high-demand applications. This complexity requires software engineers to consider factors like network latency, fault tolerance, and data consistency. Modern architectural patterns like microservices, SOA, and event-driven architecture have emerged to address these complexities and ensure system robustness. The rise of distributed databases and storage systems, such as NoSQL databases, has significantly impacted software engineering. These systems handle large data volumes, ensuring both availability and consistency across nodes. The need for robust security measures has also increased, with engineers implementing strategies to protect distributed systems from various attacks. Overall, distributed systems have driven innovation and advanced tools and methodologies, enabling the development of robust, scalable, and secure applications.

Impact of AI on Distributed Systems in Software Engineering

AI has significantly impacted distributed systems by enhancing fault tolerance, self-healing, and security. AI-driven self-healing capabilities allow systems to monitor for anomalies and predict potential failures, enabling preventative measures like rerouting traffic or restarting services, improving system resilience, and reducing downtime. AI algorithms enhance security by detecting unusual patterns and potential breaches in real time, employing techniques like anomaly detection and automated threat response.

AI optimizes data management in distributed systems by facilitating advanced data analytics and processing. It ensures efficient data distribution, indexing, and retrieval across distributed databases, providing quick access to relevant information and managing data redundancy and consistency. AI also enhances user experience by delivering personalized content and services based on user behavior and preferences.

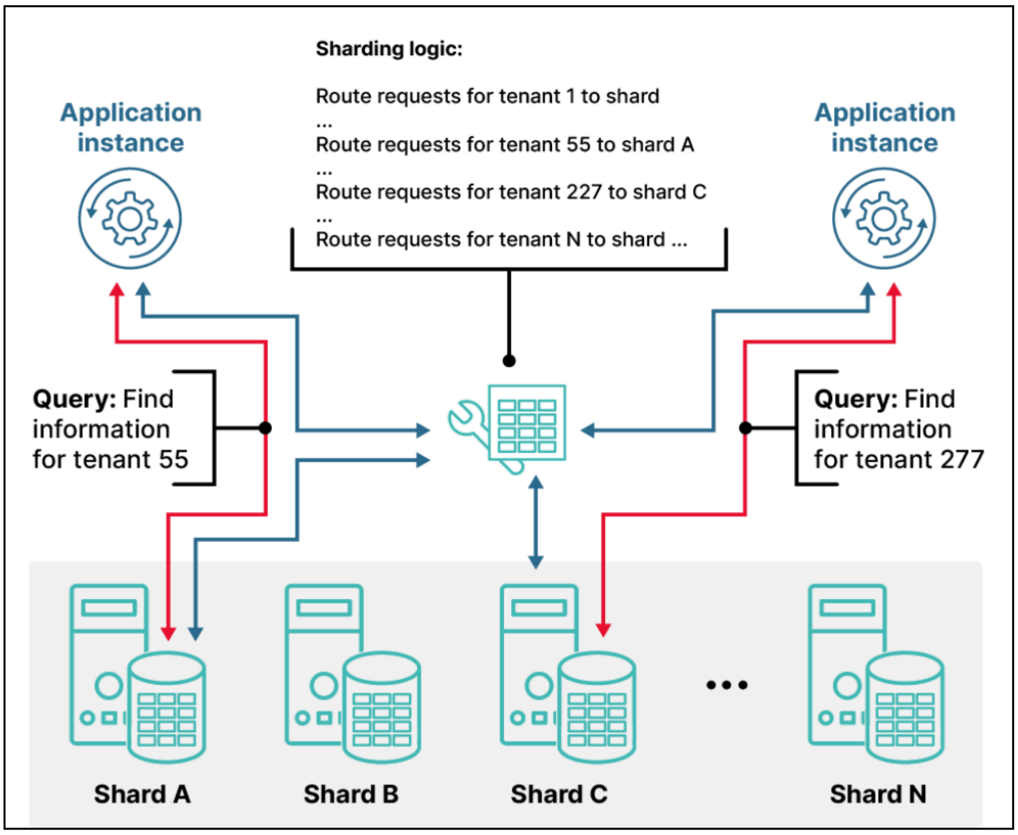
Sharding logic:
Route requests for tenant 1 to shard
...
Route requests for tenant 55 to shard A
...
Route requests for tenant 227 to shard C
...
Route requests for tenant N to shard ...
Application instance
Application instance
Query: Find information for tenant 55
Query: Find information for tenant 277
Shard A
Shard B
Shard C
Shard N

Figure 2. Illustration of the sharded pattern within a distributed system (D'Antoni, 2022)

Level of Impact

AI is a common tool used by software engineers to improve efficiency and effectiveness by automating development tasks. It has contributed to the growth of DevOps practices and CI/CD pipelines, analyzing code changes, test outcomes, and production statistics to provide insights into quality, performance, and potential issues. According to a study by researchers at the US Department of Energy's Oak Ridge National Laboratory, AI could potentially replace software developers by 2040. Additionally, AI-powered recommendation systems are widely used in e-commerce to make personalized product recommendations. Artificial intelligence is enhancing the capabilities of distributed systems, making them more advanced and adaptive. AI has advanced software development with its automation capabilities and effective resource management. By detecting potential system failures and initiating recovery processes, developers can minimize downtime and focus on more complex tasks. AI algorithms also provide intelligent recommendations by examining user data to customize user behavior which improves security and performance. A California-based market research firm specializing in software development, Evans Data Corporation, found that nearly 30% of the 550 software developers surveyed expressed fears of their roles being replaced by AI soon.

Advantages

A key advantage of AI in automation is its ability to handle tasks continuously, without the need for breaks or downtime. This constant operation ensures that processes are running around the clock, significantly speeding up project timelines. For example, AI can manage tasks such as code generation, bug detection, and system monitoring, which are often time-consuming when done manually. The uninterrupted nature of AI operations leads to a more streamlined and efficient workflow, enabling faster delivery of software products. Additionally, AI reduces time consumption and allows engineers to focus on more significant and complex aspects of their projects. By taking over mundane and repetitive activities, AI not only increases productivity but also enhances the overall quality of the software development process.

Fault tolerance is the ability of a system, whether it's hardware or software, to keep functioning correctly even if some of its components fail. Fault tolerance is important in designing strong distributed systems capable of maintaining performance even in adverse conditions. Additionally, fault tolerance in AI within distributed systems is crucial for maintaining reliable AI-driven applications and services in software engineering. One of the primary methods to achieve fault tolerance is through redundancy and replication, where AI models and their computations are distributed across multiple nodes. This redundancy ensures that if one node fails, another can take over its tasks, which is essential for scenarios such as examining autonomous systems, healthcare diagnostics, and collecting data.

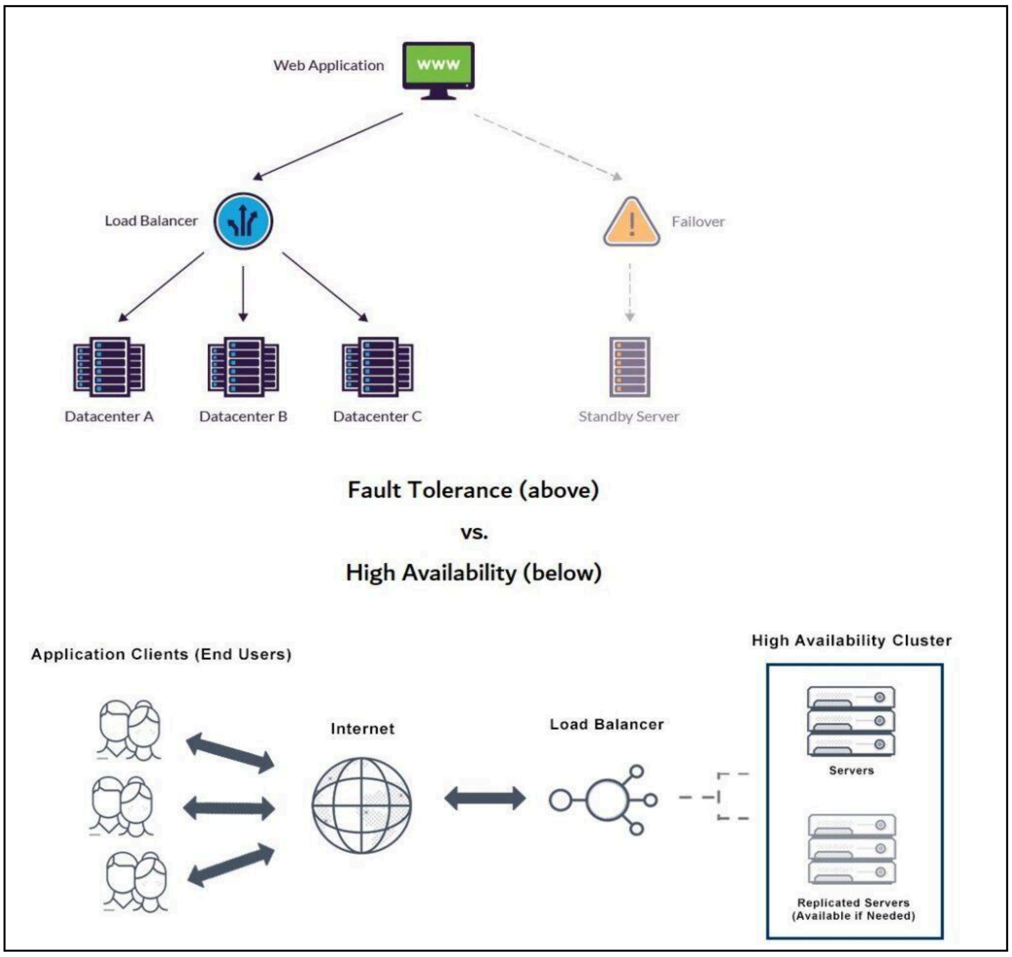

Figure 3. Fault Tolerance vs. High Availability (Imperva)

Decision Making

Artificial intelligence (AI) significantly enhances decision-making in software engineering by analyzing vast datasets and providing insights that aid engineers in making informed choices. This capability is crucial in optimizing performance, efficiently allocating resources, and refining design choices, ultimately leading to more effective software solutions. One of the core strengths of AI lies in its ability to process and analyze large volumes of data at a speed and accuracy unattainable by humans. In software engineering, this translates to the ability to quickly sift through extensive codebases, user feedback, performance metrics, and other relevant data. AI algorithms can identify patterns, detect anomalies, and uncover hidden correlations within this data, providing engineers with actionable insights. For instance, AI can analyze code repositories to identify frequently occurring bugs and suggest preventive measures, thereby reducing future errors and enhancing software reliability.

Disadvantages

Despite all the pros, AI can't replace software engineers' expertise in complex problem-solving, creativity, and innovation. The dependence on AI may also reduce creativity and critical thinking abilities among developers. Additionally, for algorithms to continue to be accurate and useful, regular maintenance, updating, and monitoring are necessary. While artificial intelligence (AI) has the potential to greatly improve distributed systems, scalability, and efficiency, it also comes with many challenges, like complexity, data privacy, and potential biases. Thus, although AI will impact software engineering, human experience remains crucial for complex system design, quality assurance, ethical considerations, and innovation. AI will help automate operations, optimize code, and help with specific tasks.

Evaluation

To evaluate whether the advantages of AI outweigh the disadvantages in distributed systems, we employed a methodology that involved selecting three key advantages: Scalability and Resource Management, Fault Tolerance and Reliability, Performance Optimization. and three significant disadvantages of AI in the context of software engineering. We conducted thorough research on the impact of each advantage and disadvantage within the field. To statistically assess whether the benefits of AI were greater than the drawbacks, we applied a two-sample t-test. This test compares the means of two independent groups to determine whether there is a statistically significant difference between them. In our case, the two groups represent the impacts of the selected advantages and disadvantages. By comparing these means, the two-sample t-test helps us understand if the observed difference is likely due to chance or if it reflects a real difference in the impact of AI's advantages and disadvantages. To determine the impact of AI on distributed systems within software engineering, we categorized the impact into three sections: low (20%), moderate (30%), and high (40%).

Based on the categories for advantages, Scalability and Resource Management was identified as having a High Impact at 40%, due to AI's critical role in efficiently managing resources and scaling operations. Fault Tolerance and Reliability was assessed with a Moderate Impact at 35%, reflecting AI's significant contribution to maintaining system stability through the prediction and prevention of failures. Lastly, Performance Optimization was categorized with a Low to Moderate Impact at 25%, indicating that while AI enhances performance, its influence is slightly less pronounced compared to scalability and reliability. Based on the categories for disadvantages, complexity and maintenance was identified as having a high Impact (40%), due to the significant increase in system complexity that AI introduces, making maintenance more challenging and resource-intensive. Security Risks was assessed with a Moderate Impact (30%), reflecting the notable vulnerabilities AI can introduce, such as data privacy concerns and potential weaknesses in algorithmic decisions. Finally, Resource Intensive was categorized with a Low Impact (20%), indicating that while AI requires substantial computational power and energy, advancements in technology can mitigate these demands to some extent. This distribution provides a clear picture of the relative disadvantages of AI, emphasizing the most significant challenges in managing AI-driven distributed systems.

Null Hypothesis (H_0): There is no difference between the average impacts of the advantages and disadvantages of AI.

Alternative Hypothesis (H_1): The average impact of the advantages is greater than the average impact of the disadvantages.

Figure 4. Inference Hypothesis

In the context of our analysis, the null hypothesis (H_0) and the alternative hypothesis (H_1) serve as two competing statements that we test using statistical methods. The null hypothesis (H_0) assumes that there is no difference between the average impacts of the advantages and disadvantages of AI in distributed systems. It suggests that any observed difference in the data is purely due to random chance or variability and that, on average, the positive and negative impacts of AI are equal. Alternative Hypothesis (H_1): This hypothesis challenges the null hypothesis by suggesting that the average impact of the advantages of AI is greater than that of the disadvantages. It implies that the benefits of AI have a more significant positive impact on distributed systems compared to the drawbacks. If the data provides enough evidence against the null hypothesis, we accept the alternative hypothesis, indicating that AI's advantages do indeed outweigh its disadvantages

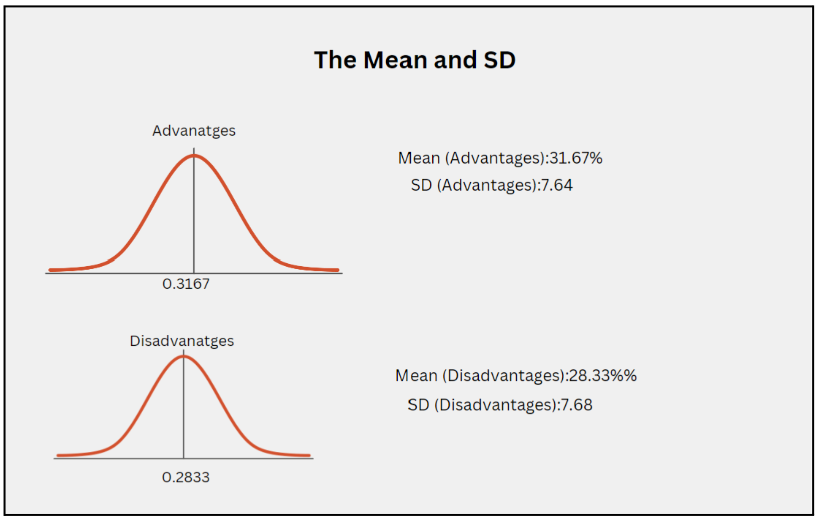

Figure 5. Normal curve distribution for advantages and disadvantages

The next step in our methodology involves comparing the mean ($\bar{x}$) impact of the advantages and disadvantages, as well as their standard deviations, to assess whether the advantages of AI in distributed systems outweigh the disadvantages. The mean, or average, impact is a measure that represents the central tendency of the data. For our analysis, the mean impact of the advantages represents the average positive effect of the selected AI advantages on distributed systems. Similarly, the mean impact of the disadvantages represents the average negative effect that the selected AI drawbacks have. By calculating these means, we obtain a summary statistic that shows us the typical impact of the advantages and disadvantages, allowing us to make a direct comparison between them. The standard deviation is a measure of the spread or variability of the data around the mean. It tells us how much the individual impact values (of either the advantages or disadvantages) deviate from their respective mean. Comparing the mean impacts allows us to determine which has a greater average effect—advantages or disadvantages, and the standard deviation helps us understand how consistent these impacts are. The mean impact for the advantages was 31.67%, with a standard deviation of 7.64, while the mean impact for the disadvantages was 28.33%, with a standard deviation of 7.68.

The next phase in our method was to use technology to determine the p-value, test statistic, and degrees of freedom, which are all important components of our statistical analysis. The test statistic, in this case the t-statistic, determines the difference between the sample averages concerning the data's variability. For our research, the t-statistic was approximately 9.58, showing a significant difference in the impacts of both advantages and disadvantages. The degrees of freedom (≈2) describe the amount of independent values that might change in the analysis, which is important when determining the critical value to compare the t-statistic against. The p-value (≈0.0054) represents the probability of observing the results under the assumption that the null hypothesis is true. Since this p-value is significantly lower than the conventional significance level of 0.05, it suggests that the benefits of AI have a statistically important impact that outweighs the disadvantages. Therefore, this supports our conclusion that the advantages of AI surpass the associated disadvantages.

Discussion

In the article The Impact of AI: How Future Software Engineers Can Adapt, Kai Yuan Neo explores how artificial intelligence is being integrated into software engineering, particularly in automating tasks such as code generation, bug detection, and testing. While AI takes over significant portions of traditional software development tasks, Neo emphasizes that human creativity, ethical decision-making, and problem-solving remain irreplaceable by AI algorithms Neo (August , 2023). Our research paper goes deeper into the possibilities of AI within software engineering and discusses its methodology, while also establishing the connection that AI has within distributed systems.

Over the years, numerous studies have evaluated the advantages and disadvantages of distributed systems in software engineering. In her article Distributed Systems Explained, Chrissy Kidd outlines how distributed systems function by using multiple computing devices across a network to manage complex tasks that a single machine could not efficiently handle. She discusses the advantages of distributed systems, such as scalability, fault tolerance, and reliability, while also highlighting challenges like increased complexity, synchronization issues, and security risks. Kidd emphasizes that understanding the architecture and operation of distributed systems is essential for software engineers, as these systems are increasingly integral to modern computing environments (Kidd, 2023)

Similarly, in What is Distributed Computing? Alexander S. Gillis provides a detailed explanation of distributed systems and the underlying frameworks supporting their use. This research also illustrates how distributed systems bring benefits to the computing world, presenting a range of system types adaptable to various applications. However, unlike our research, this paper does not explore the relations between distributed systems and AI or their role in software development.

The research study Artificial Intelligence (AI)-Centric Management of Resources in Modern Distributed Computing Systems by Shashikant Ilager, Rajeev Muralidhar, and Rajkumar Buyya explores how AI has become an important factor in modern computing environments. They explain how AI enhances the management and optimization of distributed systems through advanced techniques. One technique, known as Contemporary Distributed Computing Systems (DCS), exemplifies large-scale, complex, and heterogeneous environments that AI is designed to improve (These systems operate across multiple networks and geographical boundaries, managing vast amounts of data generated by Internet of Things (IoT)-driven applications that require real-time processing and rapid responses. Effectively managing these resources to ensure reliable service delivery is challenging, particularly with existing Resource Management Systems (RMS) that often rely on static or heuristic solutions. The advent of AI, bolstered by the availability of data and advanced processing capabilities, offers new opportunities for data-driven solutions in RMS tasks, making them more adaptive, accurate, and efficient.

Acknowledgements

This article was presented at the 13th London International Conference, held in Istanbul on 24-26 July 2024, Turkiye. It has been published in the proceedings of the London International Conferences under No. 11(2024).

References

Gillis, A. S. (n.d.). What is distributed computing? TechTarget. https://www.techtarget.com/whatis/definition/distributed-computing

Brainhub. (n.d.). Software developer in the age of AI: How to stay relevant? Brainhub. https://brainhub.eu/library/software-developer-age-of-ai#:~:text=AI%20has%20significantly%20impacted%20the,the%20DevOps%20process%20more%20efficient

Cezar, B. (2023, July 21). Distributed systems: The key to scalability, reliability, and performance. Splunk. https://www.splunk.com/en_us/blog/learn/distributed-systems.html

De Bartolomeis, J. (2024, January 3). Artificial intelligence: A threat for software engineers? Introduct.tech. https://introduct.tech/blog/artificial-intelligence-a-threat-for-software-engineers/#:~:text=Increased%20Productivity%3A%20AI%2Ddriven%20tools,cycles%20and%20higher%2Dquality%20software

Rocket Academy. (2024). The impact of AI: How future software engineers can adapt (2024 and beyond). Rocket Academy. https://www.rocketacademy.co/software-engineering/the-impact-of-ai-how-future-software-engineers-can-adapt-2024-and-beyond#:~:text=AI%20has%20revolutionised%20software%20development,focus%20on%20more%20complex%20aspects

Stanley, S. (n.d.). Distributed software engineering: Concepts and applications. Central Connecticut State University. https://cs.ccsu.edu/~stan/classes/CS530/Notes18/17-DistributedSE.html

SolarWinds. (2022, January 24). What is a distributed system? Orange Matter. https://orangematter.solarwinds.com/2022/01/24/what-is-a-distributed-system/

ScienceDirect. (n.d.). Distributed machine learning. ScienceDirect. https://www.sciencedirect.com/topics/computer-science/distributed-machine-learning

Imperva. What is fault tolerance?: Creating a fault tolerant system: Imperva. Learning Center. (2024, January 8). https://www.imperva.com/learn/availability/fault-tolerance/